



TUGAS AKHIR – TF 145565

**SISTEM MONITORING KECEPATAN PUTAR
TURBIN DAN TEGANGAN KELUARAN
GENERATOR PADA PEMBANGKIT LISTRIK
TENAGA MINI-MIKROHIDRO SKALA
LABORATORIUM**

**Delima Palwa Sari
NRP 105115.00000.005**

**Dosen Pembimbing I
Detak Yan Pratama, S.T., M.Sc.
NIP. 19840101 201212 1 002**

**Dosen Pembimbing II
Murry Raditya, S.T., M.T.
NPP. 1988 20171 1 055**

**PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018**



TUGAS AKHIR – TF145565

**SISTEM MONITORING KECEPATAN PUTAR
TURBIN DAN TEGANGAN KELUARAN
GENERATOR PADA PEMBANGKIT LISTRIK
TENAGA MINI-MIKROHIDRO SKALA
LABORATORIUM**

**Delima Palwa Sari
NRP 105115.00000.005**

**Dosen Pembimbing I
Detak Yan Pratama, S.T., M.Sc.
NIP. 19840101 201212 1 002**

**Dosen Pembimbing II
Murry Raditya, S.T., M.T.
NPP. 1988 20171 1 055**

**PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018**



FINAL PROJECT – TF145565

***MONITORING SYSTEM OF TURBINE'S
SPEED AND GENERATOR'S OUTPUT
VOLTAGE AT PLTMH LABORATORY SCALE***

**Delima Palwa Sari
NRP 105115.00000.005**

***Supervisor I*
Detak Yan Pratama, S.T., M.Sc.
NIP. 19840101 201212 1 002**

***Supervisor II*
Murry Raditya, S.T., M.T.
NPP. 1988 20171 1 055**

***DIPLOMA OF INSTRUMENTATION TECHNOLOGY
DEPARTEMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF VOCATION
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2018***

Lembar Pengesahan Tugas Akhir

SISTEM MONITORING KECEPATAN PUTAR TURBIN DAN TEGANGAN KELUARAN GENERATOR PADA PEMBANGKIT LISTRIK TENAGA MINI-MIKROHIDRO SKALA LABORATORIUM

DELIMA PALWA SARI

NRP. 1051 15 00000 005

Surabaya, Juli 2018

Dosen Pembimbing 1



Detak Yan Pratama, S.T.,M.Sc

NIP.19840101 201212 1 002

Dosen Pembimbing 2




Murry Raditya, S.T.,M.T.

NPP.1988 20171 1 055

Mengetahui,

Kepala Departemen Teknik Instrumentasi


Dr. Ir. Purwadi Agus Darwito, MSc
NIP.19620822 198803 1 001

**SISTEM MONITORING KECEPATAN PUTAR TURBIN
DAN TEGANGAN KELUARAN GENERATOR PADA
PEMBANGKIT LISTRIK TENAGA MINI-MIKROHIDRO
SKALA LABORATORIUM**




LAPORAN TUGAS AKHIR

Diberikan Untuk Memenuhi Salah Satu Syarat
Kelulusan Mata Kuliah Tugas Akhir
pada
Program Studi DIII Teknologi Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh:

DELIMA PALWA SARI
NRP. 1051 15 00000 005

Disetujui oleh pembimbing Tugas Akhir:

- | | | |
|----------------------------------|---|----------------|
| 1. Detak Yan Pratama, S.T., M.Sc |  | (Pembimbing 1) |
| 2. Murry Raditya, S.T., M.T |  | (Pembimbing 2) |
| 3. Herry Sufyan Hadi, S.T., M.T |  | (Penguji) |

SURABAYA
JULI, 2018

SISTEM MONITORING KECEPATAN PUTAR TURBIN DAN TEGANGAN KELUARAN GENERATOR PADA PLTMH SKALA LABORATORIUM

Nama Mahasiswa : DELIMA PALWA SARI
NRP : 1051 15 00000 005
Departemen : Teknik Instrumentasi FV-ITS
Dosen Pembimbing I : Detak Yan Pratama, S.T., M.Sc.
Dosen Pembimbing II : Murry Raditya, S.T., M.T.

ABSTRAK

Energi listrik merupakan faktor yang penting dalam kehidupan. Hampir seluruh aktivitas yang dilakukan manusia membutuhkan listrik. Pembangkit listrik tenaga mikrohidro (PLTMH) diciptakan untuk memenuhi kebutuhan kekurangan listrik tersebut. Pada sistem ini air dialirkan menggunakan pompa untuk menggerakkan turbin dan menghasilkan mekanik. Energi mekanik tersebut akan diubah menjadi energi listrik oleh generator. Tegangan yang diinginkan sebesar 38 volt untuk menghidupkan lampu, tetapi tegangan yang dihasilkan generator tidak selalu sesuai dengan yang diinginkan. Oleh karena itu, dirancang sebuah sistem monitoring untuk mengetahui kecepatan putar turbin serta tegangan yang dihasilkan generator. Pada perancangan sistem monitoring ini, sensor yang digunakan untuk mengukur kecepatan putar turbin adalah sensor *Proximity Hall Effect NJK500-2A*. Sedangkan sensor yang digunakan untuk mengukur tegangan keluaran generator adalah sensor Tegangan yang menggunakan modul pembagi tegangan. Dari data uji sistem monitoring didapat hasil yaitu sensor Tegangan memiliki nilai ketidakpastian diperluas (U_{exp}) sebesar ± 0.54 dengan akurasi sebesar 99.81%, dan *error* pembacaan sebesar 0.19%, sensitivitas sebesar 98,5% serta non linieritas sebesar 10.186%. Sensor *Proximity Hall Effect NJK-5002A* memiliki nilai ketidakpastian diperluas (U_{exp}) sebesar ± 12.325 dengan akurasi sebesar 95.39%, dan sensitivitas sebesar 97,56% serta non linieritas sebesar 7.49%.

Kata kunci : kecepatan putar turbin, tegangan, generator, *Proximity Hall Effect NJK500-2A* , sensor Tegangan.

**MONITORING SYSTEM OF TURBINE'S SPEED AND
GENERATOR'S OUTPUT VOLTAGE AT PLTMH
LABORATORY SCALE**

Name of Student : DELIMA PALWA SARI
NRP : 1051 15 00000 005
Departement : Departement of Instrumentation
Engineering FV-ITS
Supervisor I : Detak Yan Pratama, S.T., M.Sc.
Supervisor II : Murry Raditya, S.T., M.T.

ABSTRACT

Electrical energy is an important factor in life. Almost all human activities require electricity. Microhydro power plant (PLTMH) was created to answer the need for electricity. In this system water flowed using a pump to drive the turbine and produce mechanics energy. The mechanical energy will be converted into electrical energy by the generator. The desired voltage is 38 volts to turn on the lamp, but the generator produced the voltage does not always match with the desired one. Therefore, designed a monitoring system to determine the turbine's speed and generator's output voltage. In designing this monitoring system, the sensor used to measure the turbine's speed is Proximity Hall Effect NJK500-2A sensor. While the sensor used to measure the generator's output voltage is voltage sensor that uses voltage divider module. From the test data of the monitoring system, the result is that the Voltage sensor has an expanded uncertainty value of ± 0.54 with an accuracy of 99.81%, and error reading of 0.19%, sensitivity of 98.5% and non linearity of 10.186%. Proximity Hall Effect NJK-5002A has an expanded uncertainty value of $\pm 12,325$ with an accuracy of 95.39%, and a sensitivity of 97.56% and non linearity of 7.49%.

Keywords : Turbine's speed, generator's voltage, Proximity Hall Effect NJK500-2A, voltage sensor.

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“SISTEM MONITORING KECEPATAN PUTAR TURBIN DAN TEGANGAN KELUARAN GENERATOR PADA PLTMH SKALA LABORATORIUM”** dengan tepat waktu. terselesaikannya laporan ini juga tak luput dari dukungan dan peran serta dari orangtua dan keluarga besar serta berbagai pihak. Untuk itulah dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr. Ir. Purwadi Agus D, M.Sc. selaku Kepala Departemen Teknik Instrumentasi FV-ITS.
2. Bapak Detak Yan Pratama, ST, M.Sc. dan Murry Raditya ST., M.T selaku pembimbing Tugas Akhir yang telah membina dengan baik dan sabar.
3. Bapak Dr. Ir. Ali Musyafa', M.Sc. selaku Dosen Wali penulis.
4. Kedua orang tua dan adik saya yang tidak henti-hentinya memberi semangat dan doa untuk menyelesaikan Tugas Akhir ini
5. PLTMH *team* yang telah membantu pengerjaan Tugas Akhir ini hingga selesai.
6. Teman-teman D3 Teknik Instrumentasi angkatan 2015 FV-ITS.
7. Serta semua pihak yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih kurang sempurna. Oleh karena itu penulis menerima segala masukan baik berupa saran maupun kritik demi kesempurnaan laporan ini.

Demikian laporan Tugas Akhir ini penulis persembahkan dengan harapan dapat bermanfaat dalam akademik baik bagi penulis sendiri maupun bagi pembaca.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

| | Hal |
|-------------------------------|-------------|
| HALAMAN JUDUL..... | i |
| LEMBAR PENGESAHAN..... | iii |
| ABSTRAK..... | v |
| ABSTRACT | vi |
| KATA PENGANTAR | vii |
| DAFTAR ISI..... | viii |
| DAFTAR GAMBAR..... | xi |
| DAFTAR TABEL..... | xiii |

BAB I. PENDAHULUAN

| | |
|---------------------------|---|
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Tujuan | 2 |
| 1.4 Batasan Masalah..... | 2 |
| 1.5 Manfaat..... | 3 |

BAB II. DASAR TEORI

| | |
|---|----|
| 2.1 <i>Micro-Hydro</i> | 5 |
| 2.2 Sistem Monitoring..... | 5 |
| 2.3 Turbin Pelton..... | 6 |
| 2.4 HMI (Human Machine Interface)..... | 7 |
| 2.5 Sensor Kecepatan Putar Turbin..... | 8 |
| 2.6 Sensor Tegangan | 9 |
| 2.7 Mikrokontroler ATmega..... | 10 |
| 2.8 <i>Data Logger</i> | 13 |
| 2.9 Indikator | 14 |
| 2.10 <i>Display</i> | 15 |
| 2.11 <i>Analog to Digital Converter (ADC)</i> | 16 |
| 2.12 CodeVisionAVR | 17 |
| 2.13 Khazama..... | 18 |
| 2.14 Karakteristik statik | 18 |
| 2.15 Teori ketidakpastian | 20 |

BAB III. PERANCANGAN DAN PEMBUATAN ALAT

| | |
|--|----|
| 3.1 Diagram Alir (<i>Flowchart</i>) | 23 |
| 3.2 Identifikasi Sistem Monitoring dan Alat Ukur Tegangan serta RPM | 26 |
| 3.3 Pemodelan <i>Hardware</i> dan <i>Software</i> | 26 |
| 3.4 Pembuatan <i>Software</i> dan <i>Hardware</i> | 26 |
| 3.5 Integrasi <i>Software</i> dan <i>Hardware</i> Pemrograman | 39 |
| 3.6 Pengujian Sistem Pengukuran | 39 |
| 3.7 Pembuatan Desain HMI pada Visual Basic dan Data Logger..... | 40 |
| 3.8 Integrasi <i>Software</i> , <i>Hardware</i> dan Sistem Monitoring | 43 |
| 3.9 Pengambilan Data Karakteristik Statik dan Kalibrasi | 43 |
| 3.10 Analisa Data dan Penarikan Kesimpulan | 43 |

BAB IV. PENGUJIAN ALAT DAN ANALISIS DATA

| | |
|--|----|
| 4.1 Pengujian Alat | 45 |
| 4.1.1 Realisasi Pemasangan Sensor | 46 |
| 4.1.2 Pemrosesan Data | 46 |
| 4.1.3 Data Logger..... | 48 |
| 4.1.4 Tampilan LCD..... | 49 |
| 4.1.5 HMI | 50 |
| 4.2 Hasil Uji Karakteristik Alat..... | 55 |
| 4.2.1 Karakteristik Statis Sensor Tegangan..... | 55 |
| 4.2.2 Kalibrasi Sensor Tegangan..... | 57 |
| 4.2.3 Karakteristik Statis Sensor Proximity Hall Effect NJK-5002A | 59 |
| 4.2.4 Kalibrasi Sensor Proximity Hall Effect NJK-5002A .. | 61 |
| 4.3 Pembahasan..... | 62 |

BAB V. PENUTUP

| | |
|---------------------|----|
| 5.1 Kesimpulan..... | 65 |
| 5.2 Saran..... | 65 |

DAFTAR PUSTAKA

LAMPIRAN A (*DATA SHEET SYSTEM ATMEGA128*)

**LAMPIRAN B (*Data Sheet Proximity Hall Effect sensor
NJK500-2A*)**

LAMPIRAN C (LISTING PROGRAM)

LAMPIRAN D (DATA KALIBRASI SENSOR TEGANGAN)

**LAMPIRAN E (DATA KALIBRASI SENSOR PROXIMITY
HALL EFFECT)**

LAMPIRAN F (*DATA SHEET* SENSOR TEGANGAN)

BIODATA PENULIS

DAFTAR GAMBAR

| | Hal |
|---|-----|
| Gambar 2.1 Sudu Turbin Pelton | 7 |
| Gambar 2.2 Sensor RPM Magnetik (<i>Proximity Hall Effect</i>) ... | 8 |
| Gambar 2.3 Prinsip Kerja Sensor RPM Magnetik (<i>Hall Effect</i>)..... | 9 |
| Gambar 2.4 Sensor Tegangan | 10 |
| Gambar 2.5 <i>Board</i> Mikrokontroler ATmega 128 | 11 |
| Gambar 2.6 <i>Chip</i> Mikrokontroler ATmega 128..... | 12 |
| Gambar 2.7 Konfigurasi Pin Mikrokontroler ATmega128 ... | 13 |
| Gambar 2.8 Modul OpenLogger | 14 |
| Gambar 2.9 Lampu 5W12V | 14 |
| Gambar 2.10 Diagram Blok LCD 20 × 4 cm..... | 15 |
| Gambar 2.11 Proses konversi sinyal analog menjadi sinyal digital | 17 |
| Gambar 2.12 Tampilan CodeVision AVR..... | 17 |
| Gambar 2.13 Khazama V 1.7.0..... | 18 |
| Gambar 2.14 Tabel T-Student | 21 |
| Gambar 3.1 <i>Flowchart</i> Tugas Akhir | 23 |
| Gambar 3.2 Desain Desain mini <i>plant</i> PLTMH | 24 |
| Gambar 3.3 PFD dan P&ID mini <i>plant</i> PLTMH..... | 25 |
| Gambar 3.4 Pemodelan Sistem Tampak Samping dan Atas... | 26 |
| Gambar 3.5 Peletakan sensor RPM (a) dan sensor tegangan (b) pada <i>plant</i> PLTMH..... | 27 |
| Gambar 3.6 Diagram blok sistem monitoring kecepatan turbin dan tegangan keluaran generator pada PLTMH.. | 27 |
| Gambar 3.7 Skematik Rangkaian Modul Sensor Tegangan pada <i>Software</i> EAGLE..... | 28 |
| Gambar 3.8 Modul Sensor Tegangan | 29 |
| Gambar 3.9 Mini <i>Plant</i> PLTMH | 29 |
| Gambar 3.10 Buka <i>Software</i> | 30 |
| Gambar 3.11 <i>Create New File</i> | 31 |
| Gambar 3.12 <i>Chip</i> pada <i>Code Vision AVR</i> | 31 |
| Gambar 3.13 <i>Alphanumeric LCD</i> pada <i>Code Vision AVR</i> | 32 |
| Gambar 3.14 Tombol <i>Generate</i> | 32 |

| | | |
|--------------------|--|----|
| Gambar 3.15 | Tampilan Awal Program..... | 33 |
| Gambar 3.16 | Flowchart HMI..... | 41 |
| Gambar 3.17 | Flowchart proses penyimpanan data pada Open Logger | 42 |
| Gambar 4.1 | Diagram blok sistem monitoring kecepatan turbin dan <i>voltage</i> generator pada PLTMH | 45 |
| Gambar 4.2 | Realisasi Pemasangan Sensor RPM (A) dan Sensor Tegangan (B) | 46 |
| Gambar 4.3 | <i>Data Processing</i> Sensor Kecepatan Putar Turbin..... | 46 |
| Gambar 4.4 | <i>Data Processing</i> Sensor Tegangan | 47 |
| Gambar 4.5 | Tampilan hasil penyimpanan Data Logger menggunakan Modul OpenLog..... | 48 |
| Gambar 4.6 | Tampilan LCD pada Panel..... | 49 |
| Gambar 4.7 | Tampilan menu utama HMI..... | 50 |
| Gambar 4.8 | Tampilan untuk tab Connection..... | 51 |
| Gambar 4.9 | Tampilan untuk tab Grafik..... | 52 |
| Gambar 4.10 | Tampilan untuk tab Monitoring Kecepatan Turbin dan Tegangan | 53 |
| Gambar 4.11 | Tampilan untuk tab Database..... | 54 |
| Gambar 4.12 | Grafik Pembacaan Sensor Tegangan | 56 |
| Gambar 4.13 | Grafik Linieritas Sensor Tegangan | 57 |
| Gambar 4.14 | Grafik Pembacaan Sensor Proximity Hall Effect NJK500-2A | 60 |
| Gambar 4.15 | Grafik Linieritas Sensor Proximity Hall Effect NJK500-2A..... | 61 |

DAFTAR TABEL

| | Hal |
|--|-----|
| Tabel 2.1 Konfigurasi Pin LCD 20×4 cm..... | 15 |
| Tabel 4.1 Data Sensor Tegangan | 55 |
| Tabel 4.2 Data Pengujian Sensor Proximity Hall Effect | 59 |

BAB I

PENDAHULUAN

1.1. Latar Belakang

Energi merupakan salah satu kebutuhan yang mutlak yang harus dipenuhi dalam kehidupan manusia, baik itu energi yang diperoleh dari alam maupun energi yang diperoleh dari proses manipulasi pemanfaatan energi oleh manusia. Terutama energi listrik, listrik merupakan faktor yang penting dalam kehidupan, baik pada sektor rumah tangga, penerangan, komunikasi industri dan sebagainya. Hampir seluruh aktifitas yang dilakukan manusia membutuhkan listrik. Dewasa ini pembangkit listrik tenaga uap (PLTU) sudah bisa tergantikan oleh berbagai energi terbarukan, salah satunya adalah PLTA atau pembangkit listrik tenaga air, yang prinsipnya menggunakan air sebagai tenaga penggerak dengan cara memanfaatkan tinggi terjunan dan jumlah debit air per detik yang dipicu dengan pompa sehingga akan memutar poros turbin dan menghasilkan energi mekanik. Energi ini selanjutnya menggerakkan generator dan menghasilkan listrik (Wirabudi, 2016).

Pada proses pembuatan sebuah sistem pembangkit listrik tenaga air (PLTA) dibutuhkan luas area yang sangat luas, maka dari itu diciptakanlah sebuah pemanfaatan energi yaitu berupa pembangkit listrik tenaga mikrohidro (PLTMH) yang dapat menghasilkan suatu produk energi berupa tenaga listrik. Mikrohidro merupakan sebuah istilah yang terdiri dari kata *micro* yang berarti kecil dan *hydro* yang berarti air. Secara teknis, mikrohidro memiliki tiga komponen utama yaitu air (sumber energi), turbin dan generator. Pada penelitian Mikrohidro ini untuk mendapatkan energi dari aliran air yaitu dengan memanfaatkan energi potensial dan energi kinetik air sebagai penggerak turbin, dimana konsep aliran air yang dipakai digantikan dengan menggunakan pompa air. Kemudian energi mekanik yang berasal dari putaran poros turbin akan diubah menjadi energi listrik oleh sebuah generator. Untuk peningkatan

hasil energi listrik maka diperlukan suatu perangkat atau alat yang dapat digunakan untuk mendukung proses tersebut (WingWijayanto, 2012).

Oleh karena itu, dalam rangka pengerjaan tugas akhir mahasiswa dirancang sebuah sistem monitoring kecepatan turbin dan tegangan keluaran generator untuk mengetahui kecepatan aliran arus air, kecepatan putar turbin generator serta tegangan yang dihasilkan pada *mini-microhidro* tersebut berbasis mikrokontroler AVR Atmega.

1.2. Rumusan Masalah

Adapun permasalahan yang akan diselesaikan adalah bagaimana merancang sistem monitoring kecepatan turbin dan tegangan keluaran generator pada pembangkit listrik tenaga mini-mikrohidro skala laboratorium yang berbasis mikrokontroler AVR ATmega sehingga diketahui kecepatan turbin generator dan tegangan yang dihasilkan oleh generator tersebut.

1.3. Tujuan

Adapun tujuan dari tugas akhir ini adalah untuk merancang dan membuat sistem monitoring turbin dan generator pada *mini-microhidro* skala laboratorium berbasis mikrokontroler AVR ATmega, yang digunakan untuk mengetahui kecepatan putaran dari turbin generator tersebut dan tegangan yang dihasilkan oleh generator tersebut.

1.4. Batasan Masalah

Adapun batasan masalah dari sistem yang dirancang ini adalah :

1. Alat yang dirancang dan diimplementasikan berfungsi untuk mengetahui kecepatan aliran arus air, kecepatan putar turbin generator serta daya dan tegangan yang dihasilkan pembangkit listrik tenaga mikro-hidro (PLTMH).
2. Data input dari sensor dan alat ukur berupa satu buah sensor kecepatan putaran turbin *rpm magnetic sensor (hall effect sensor)* dan satu buah sensor tegangan yang telah diolah pada

mikrokontroler AVR ATmega sistem pengendalian masing-masing.

3. Data input tersebut kemudian dikirimkan ke komputer menggunakan modul selanjutnya dimasukkan pada *data logger*.

1.5. Manfaat

Adapun manfaat yang diperoleh dari tugas akhir ini yaitu :

1. Tugas akhir ini dapat dijadikan bekal bagi peserta untuk kedepannya dalam menghadapi dunia industri yang berkaitan dengan sistem kontrol.
2. Tugas akhir ini dapat dijadikan sebagai simulasi dalam praktikum beberapa mata kuliah yang ada di departemen.

Halaman ini sengaja dikosongkan

BAB II

DASAR TEORI

2.1. *Micro-Hydro*

Microhydro merupakan sebuah istilah yang terdiri dari kata *micro* yang berarti kecil dan *hidro* yang berarti air. Secara teknis, *microhydro* memiliki tiga komponen utama yaitu air (sumber energi), turbin dan generator. Air merupakan sumber *energy* sebagai penggerak turbin. Turbin merupakan sejenis kipas yang akan berputar apabila mendapatkan energi, misalnya energi air. Sedangkan generator merupakan suatu alat yang dapat merubah energi gerak menjadi energi listrik.

Microhydro mendapatkan energi dari aliran air yang memiliki perbedaan ketinggian tertentu. Pada dasarnya, mikrohidro memanfaatkan energi potensial jatuhnya air (*head*). Semakin tinggi jatuhnya air maka semakin besar energi potensial air yang dapat diubah menjadi energi listrik. Di samping faktor geografis (tata letak sungai), tinggi jatuhnya air dapat pula diperoleh dengan membendung aliran air sehingga permukaan air menjadi tinggi. Air dialirkan melalui sebuah pipa pesat kedalam rumah pembangkit yang pada umumnya dibangun di bagian tepi sungai untuk menggerakkan turbin atau kincir air *microhydro*. Kemudian energi mekanik yang berasal dari putaran poros turbin akan diubah menjadi energi listrik oleh sebuah generator (WingWijayanto, 2012).

2.2. Sistem Monitoring

Monitoring adalah pemantauan yang dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari itu. *Monitoring* akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, misalnya untuk memeriksa terhadap suatu proses atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil sebuah sistem atas efek tindakan dari beberapa jenis antara lain tindakan

untuk mempertahankan suatu sistem yang sedang berjalan. Berdasarkan judul diatas sistem monitoring bertujuan untuk memantau bagaimana perubahan nilai besaran fisis yang ada pada plant. Sehingga data dapat dijadikan acuan yang tepat.

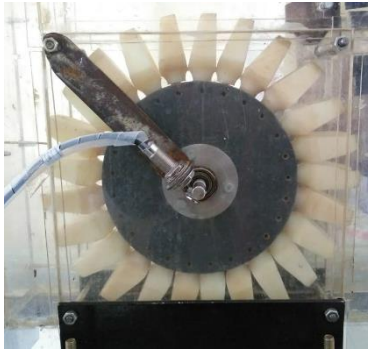
Sistem monitoring terdiri dari beberapa proses yaitu sebagai berikut:

1. *Sensing* : yaitu untuk mengukur besaran fisis yang di inginkan seperti tegangan atau kecepatan putar turbin, pada umumnya untuk mengukur besaran fisis digunakan sensor/transduser yang akan mengubah besaran fisis menjadi besaran listrik.
2. *Prosesing* : yaitu proses mengubah data analog menjadi digital atau pada umumnya disebut ADC(analog to digital converter) mikrokontroler berperan penting dalam menjalankan ADC ini.
3. *Recording* : adalah perekaman data yang telah dibaca agar kita dapat menampilkan data tersebut pada *display*.
4. *Display* : merupakan proses menampilkan data yang telah diolah oleh controller dengan tujuan agar besaran fisis dapat dengan mudah dibaca.
5. *Saving data* : merupakan proses terakhir yaitu proses penyimpanan data yang telah dibaca agar kita dapat menjadikan acuan apabila ingin mengolah data tersebut

2.3. Turbin Pelton

Turbin pelton merupakan turbin impuls yang prinsip kerjanya mengubah energi potensial air menjadi energi kinetik dalam bentuk pancaran air. Pancaran air yang keluar dari mulut nozel diterima oleh mangkok-mangkok pada roda jalan sehingga roda jalan berputar. Dari putaran inilah menghasilkan energi mekanik yang memutar poros generator sehingga menghasilkan energi listrik. Perancangan turbin pelton dimulai dengan melakukan penegenal dan identifikasi, pertimbangan dan melihat faktor keamanannya untuk dapat menyajikan hasil rancangan dengan baik begitu juga dengan material yang akan digunakan dalam perancangan turbin pelton. Dalam perancangan di butuhkan massa jenis atau density (ρ) untuk dapat menentukan berat jenisnya, massa jenis merupakan massa suatu benda per satuan volume. Sedangkan berat jenis benda

adalah berat suatu benda dalam satuan volume, berat mempunyai arah, berat suatu benda dipengaruhi oleh massa benda dan gravitasi.



Gambar 2.1 Sudu Turbin Pelton

Adapun cara kerja turbin pelton ini adalah Tahap pertama yang dilakukan adalah menghidupkan saklar motor pompa yang terdapat pada panel listrik yang bertujuan agar pompa berputar dan menghisap air yang berada pada bak penampung, lalu air mengalir melalui pipa menuju ke nozzle penyemprot. Sebelum sampai ke nozzle air terlebih dahulu melewati kran dan juga alat ukur flow meter. Dan dari nozzle lalu air disemprotkan ke sudu turbin yang menyebabkan turbin serta As turbin berputar sehingga kita bisa mengukur prestasi turbin. Dan air yang disemprotkan oleh nozzle ke sudu itu jatuh kembali pada bak penampungan air (Dr. Sri Purnomo, 2001).

2.4. HMI (Human Machine Interface)

HMI (Human Machine Interface) adalah sebuah interface atau tampilan penghubung antara manusia dengan mesin. HMI juga merupakan user interface dan sistem kontrol untuk manufaktur. HMI mempunyai fungsi sebagai berikut :

- Memonitor keadaan yang ada di *plant*.
- Mengambil tindakan yang sesuai dengan keadaan yang terjadi.

- Memunculkan tanda peringatan dengan menggunakan alarm jika terjadi sesuatu yang tidak normal.
- Menampilkan pola data kejadian yang ada di *plant* baik real time.

HMI bertugas untuk memvisualisasikan kejadian, peristiwa, atau pun proses yang sedang terjadi di *plant* secara nyata sehingga dengan HMI, operator lebih mudah dalam melakukan pekerjaan fisik (Prastowo, 2015). Biasanya HMI digunakan juga untuk menunjukkan kesalahan mesin, status mesin, memudahkan operator untuk memonitor beberapa part pada plant. Plant tracking ini memvisualkan hasil monitoring ke sebuah HMI agar manusia dapat memahami bahasa mesin dengan mudah.

2.5. Sensor RPM

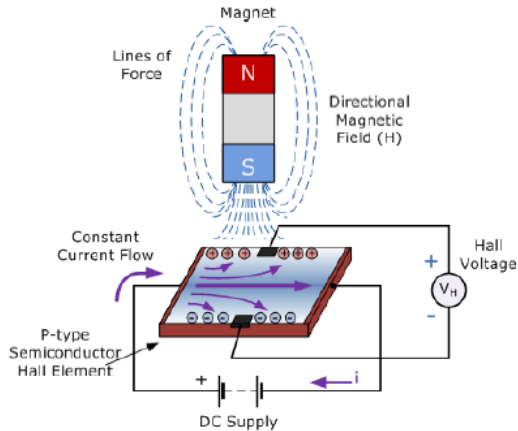
Proximity Hall Effect sensor atau sensor *proximity* medan magnet adalah sensor yang berfungsi untuk mendeteksi medan magnet. *Proximity Hall Effect sensor* memberikan output berupa tegangan yang proporsional dengan kekuatan medan magnet yang diterima oleh sensor tersebut. Sensor *Proximity Hall Effect* ini dibangun dari sebuah lapisan silikon dan dua buah elektroda pada masing-masing sisi silikon (Texas Instrument , 2014).



Gambar 2.2 Sensor RPM Magnetik (*Proximity Hall Effect*)

Pada dasarnya, sensor ini mendeteksi tarikan gaya magnet di sekitar penampang sensor. Pada saat perangkat didekatkan dengan

medan magnet, maka garis fluks magnet mengakibatkan penyebaran muatan beban serta muatan electron, sehingga terjadi perubahan tegangan keluaran dari sensor magnet (Texas Instrument , 2014).



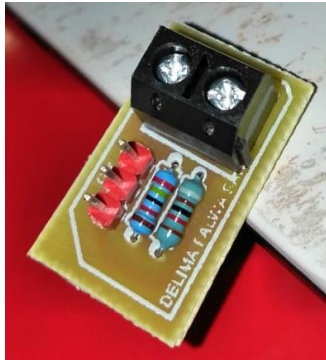
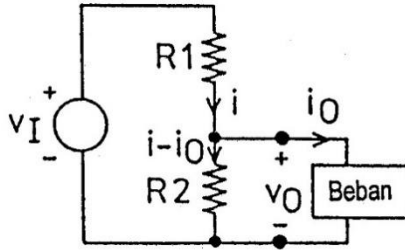
Gambar 2.3 Prinsip Kerja Sensor RPM Magnetik (*Hall Effect*)

2.6. Sensor Tegangan

Sensor tegangan yang digunakan merupakan modul yang menggunakan prinsip pembagi tegangan. Modul ini dapat mengurangi tegangan input hingga 20 kali dari tegangan asli. Tegangan analog input maksimum mikrokontroler yaitu 5 volt, sehingga modul tegangan dapat diberi masukkan tidak melebihi 20×5 Volt atau sebesar 100 Volt. Untuk dapat mengukur tegangan dengan benar maka digunakan rumus:

$$\text{Tegangan} = \text{ADC} \times 100 / \text{bitkontroler};$$

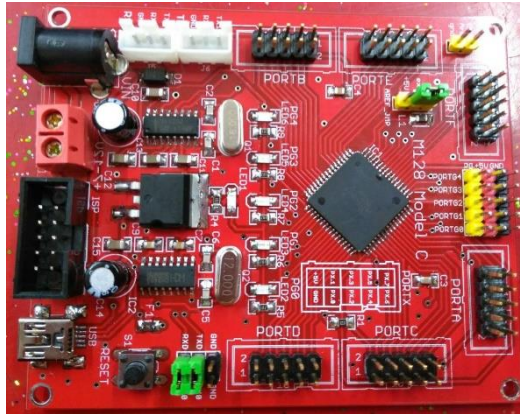
Dimana ADC adalah data pembacaan sinyal analog dari plant dan akan dibagi oleh V_{reff} dari kontroller, bergantung dengan jenis kontrollernya maka nilai V_{reff} juga akan berbeda-beda. Pada tugas akhir ini V_{reff} yang digunakan sebesar 5volt dan 8 bit yaitu 255.



Gambar 2.4 Sensor Tegangan

2.7. Mikrokontroler ATmega

Mikrokontroler ATmega128 merupakan salah satu varian dari mikrokontroler AVR 8-bit. Beberapa fitur yang dimiliki adalah memiliki beberapa memory yang bersifat non-volatile, yaitu 128Kbytes of In-System Self-Programmable Flash program memory (128Kbytes memory flash untuk pemrograman), 4Kbytes memori EEPROM, 4Kbytes memori Internal SRAM, write/erase cycles : 10.000 Flash/ 100.000 EEPROM (program dalam mikrokontroler dapat diisi dan dihapus berulang kali sampai 10.000 kali untuk flash memori atau 100.000 kali untuk penyimpanan program/data di EEPROM).



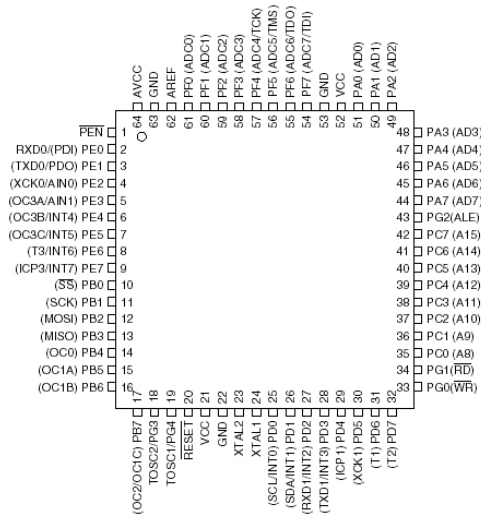
Gambar 2.5 Board Mikrokontroler ATmega 128

Selain memori, fitur yang dimiliki oleh mikrokontroler atmega128 ini adalah pada perangkat peripheral interfacenya, yaitu memiliki 2 buah 8-bit Timer/Counter, 2 buah expand 16-bit Timer/Counter, RTC (Real Time Counter) dengan oscillator yang terpisah, 2 buah 8-bit chanel PWM, 6 PWM channel dengan resolusi pemrograman dari 2 sampai 16 bits, output compare modulator, 8-channel 10-bit ADC, 2 buah TWI (Two Wire Interface), 2 buah serial USARTs, Master/Slave SPI serial interface, Programmable Watchdog Timer dengan On-chip Oscillator, On-chip analog comparator, dan memiliki 53 programmable I/O. Sedangkan untuk pengoperasiannya sendiri, Mikrokontroler ATmega128 dapat dioperasikan pada catuan 2.7 – 5.5 V untuk ATmega128L (low voltage) dengan clock speed 0 – 8 MHz dan 4.5 – 5.5 V untuk ATmega128 dengan clock speed 0 – 16 MHz.



Gambar 2.6 *Chip* Mikrokontroler ATmega 128

Sistem minimum merupakan suatu rangkaian minimalis yang dirancang / dibuat agar suatu mikrokontroler dapat berfungsi dan bekerja dengan semestinya. Sama seperti mikrokontroler atmega8535, atmega128 juga membutuhkan sistem minimum, Namun sistem minimum pada Mikrokontroler ATmega128 memiliki beberapa perbedaan dibandingkan dengan sistem minimum mikrokontroler keluarga AVR yang lain. Perbedaan terletak pada konfigurasi pin pada ISP (In System Programming). Jika pada kebanyakan mikrokontroler jenis AVR konfigurasi pin untuk ISP-nya adalah mosi-mosi, miso-miso, sck-sck, reset-reset, dan power supply, maka pada Mikrokontroler ATmega128 adalah mosi-RX0, miso-TX0, SCK-SCK, dan power supply. Berikut adalah konfigurasi pin Mikrokontroler ATmega128 :



Gambar 2.7 Arsitektur Mikrokontroler ATmega 128

2.8. Data Logger

Data logger (perekam data) adalah sebuah alat elektronik yang mencatat data dari waktu ke waktu baik yang terintegrasi dengan sensor dan instrumen didalamnya maupun eksternal sensor dan instrumen. Atau secara singkat *data logger* adalah alat untuk melakukan *data logging*. *Logging* data (*data logging*) adalah proses otomatis pengumpulan dan perekaman data dari sensor untuk tujuan pengarsipan atau tujuan analisis. Sensor digunakan untuk mengkonversi besaran fisik menjadi sinyal listrik yang dapat diukur secara otomatis dan akhirnya dikirimkan ke komputer atau mikroprosesor untuk pengolahan. Berbagai macam sensor sekarang tersedia. Selain itu, banyak peralatan laboratorium dengan *output* listrik dapat digunakan bersama dengan konektor yang sesuai dengan *data logger* (Hartono, 2013)



Gambar 2.8 Modul OpenLogger

2.9. Indikator

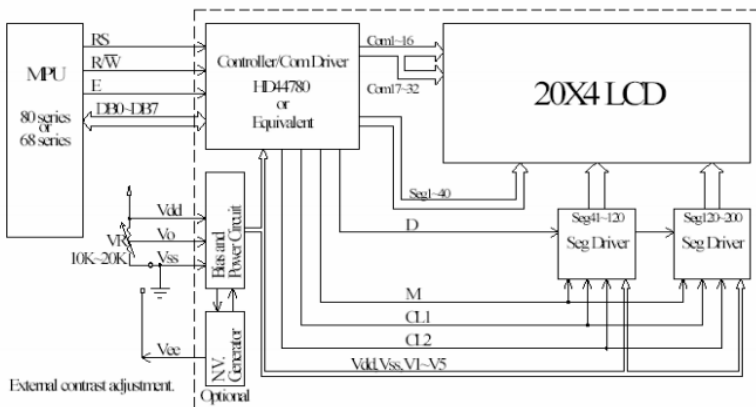
Pada *plant* PLTMH ini indikator yang digunakan untuk mengetahui adanya listrik yang dihasilkan yaitu menggunakan lampu 12V 9W sebagai media cahaya yang menyala karena adanya aliran listrik hasil dari putaran turbin yang ditransmisiikan ke generator sehingga menghasilkan daya listrik.



Gambar 2.9 Lampu 9W12V

2.10. Display

Pada sistem monitoring *display* yang digunakan pada plant yaitu LCD (*liquid Crystal Display*) merupakan suatu perangkat elektronika yang telah terkonfigurasi dengan kristal cair dalam gelas plastik atau kaca sehingga mampu memberikan tampilan berupa titik, garis, simbol, huruf, angka ataupun gambar. LCD terbagi menjadi dua macam berdasarkan bentuk tampilannya, yaitu *Text-LCD* dan *Graphic-LCD*. Berupa huruf atau angka, sedangkan bentuk tampilan pada *Graphic-LCD* berupa titik, garis dan gambar. Dalam LCD setiap karakter ditampilkan dalam matriks 5x7 pixel. Pada gambar di bawah ini merupakan LCD 4 x 20 yang berguna untuk menampilkan pembacaan sensor rpm, *flowrate*, dan tegangan yang sudah di olah di mikrokontroler dan kemudian ditampilkan ke LCD untuk menjadi *interface* hasil pembacaan sensor (Nurcahyo , 2012).



Gambar 2.10 Diagram Blok LCD 20 × 4 cm

Berikut adalah konfigurasi kaki-kaki LCD karakter 20x4.

Tabel 2.1 Konfigurasi Pin LCD 20 × 4 cm.

| PIN NO. | SYMBOL | FUNCTION |
|---------|-----------------|----------------|
| 1 | V _{SS} | Ground |
| 2 | V _{DD} | + 3 V or + 5 V |

| | | |
|----|-----------------|-------------------------------|
| 3 | V_0 | Contrast adjustment |
| 4 | RS H/L | register select signal |
| 5 | R/W H/L | read/write signal |
| 6 | E H \square L | enable signal |
| 7 | DB0 H/L | data bus line |
| 8 | DB1 H/L | data bus line |
| 9 | DB2 H/L | data bus line |
| 10 | DB3 H/L | data bus line |
| 11 | DB4 H/L | data bus line |
| 12 | DB5 H/L | data bus line |
| 13 | DB6 H/L | data bus line |
| 14 | DB7 H/L | data bus line |
| 15 | A | Power supply for LED (4.2 V) |
| 16 | K | Power supply for B/L (0 V) |
| 17 | NC/ V_{EE} | NC or negative voltage output |
| 18 | NC | NC connection |

2.11. *Analog to Digital Conversion (ADC)*

Mendapatkan informasi dari sebuah sinyal menggunakan perangkat analog adalah rumit dan kurang akurat. Karena itu kita gunakan metode pengolahan yang lebih sederhana, fleksibel dan akurat, yaitu pengolahan sinyal digital (DSP).

Untuk pengolah sinyal analog dengan perangkat digital, yang pertama dilakukan adalah mengubah sinyal analog menjadi sederetan angka yang mempunyai keakuratan tertentu. Langkah ini disebut konversi analog ke digital, menggunakan alat yang disebut ADC (Analog to Digital Converter). Supaya sinyal digital ini cukup akurat untuk dikembalikan lagi menjadi sinyal analog maka perlu diperhatikan masalah jumlah sampling yang dipilih oleh ADC dan besarnya angka yang dipakai untuk mewakili tiap sampling. Teori sampling membantu kita untuk menentukan jumlah sampling yang diperlukan untuk menghasilkan kembali sinyal analog berdasarkan frekuensi maksimum pada sinyal analog yang diolah. Blok diagram dasar dari sebuah ADC ditunjukkan oleh gambar 2.9 (PENS, 2018).

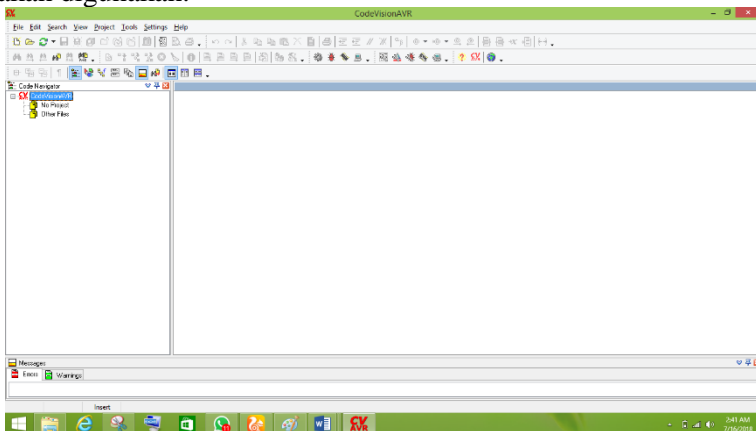


Gambar 2.11 Proses konversi sinyal analog menjadi sinyal digital

2.12. CodeVision AVR

CodeVision AVR merupakan sebuah *software* yang digunakan untuk memprogram mikrokontroler sekarang ini telah umum. Mulai dari penggunaan untuk kontrol sederhana sampai kontrol yang cukup kompleks, mikrokontroler dapat berfungsi jika telah diisi sebuah program, pengisian program ini dapat dilakukan menggunakan compiler yang selanjutnya diprogram ke dalam mikrokontroler menggunakan fasilitas yang sudah disediakan oleh program tersebut. Salah satu *compiler* program yang umum digunakan sekarang ini adalah CodeVision AVR yang menggunakan bahasa pemrograman C.

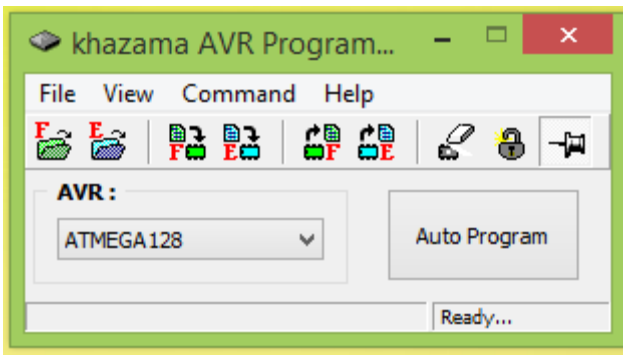
CodeVision AVR mempunyai suatu keunggulan dari compiler lain, yaitu adanya *codewizard*, fasilitas ini memudahkan pengguna atau pemrogram dalam inisialisasi mikrokontroler yang akan digunakan.



Gambar 2.12 Tampilan CodeVision AVR

2.13. Khazama

Software ini digunakan untuk meng-*upload listing* program yang dibuat pada *code vision AVR* ke mikrokontroler atmega128. Selain itu, khazama v 1.7.0 juga dapat melakukan pengaturan *fuse bit* dan proses eksekusi program pada mikrokontroler. Tampilan khazama v 1.7.0 yang digunakan pada tugas akhir kali ini dapat dilihat pada gambar 2.14.



Gambar 2.13 Khazama V 1.7.0

2.14. Karakteristik Statik

Karakteristik statik pengukuran merupakan karakteristik yang ditentukan melalui perhitungan matematik atau secara grafik. Karakteristik statik merupakan karakter yang menggambarkan parameter dari sebuah instrument pada saat keadaan *steady* (Anonim, 2016). Karakteristik statik terdiri dari:

a. Range

Range merupakan selisih antara nilai minimum dan maksimum yang terukur oleh suatu instrument atau alat ukur.

b. Akurasi

Akurasi merupakan tingkat ketelitian suatu alat dalam memberikan hasil pengukuran. Akurasi diperoleh dari persamaan berikut:

$$A = 1 - \left| \frac{Y_n - X_n}{Y_n} \right| \times 100\% \quad 2.1$$

dimana :

Y_n = pembacaan validator

X_n = pembacaan alat

c. Toleransi

Toleransi menunjukkan kesalahan maksimum yang diperbolehkan pada hasil pembacaan alat ukur.

d. Sensitivitas

Sensitivitas menunjukkan perubahan output instrument yang terjadi saat diberi kualitas pengukuran yang berbeda. Sensitivitas diperoleh dari persamaan berikut:

$$S = \Delta O / \Delta I \quad 2.2$$

dimana O adalah output dan I adalah input.

e. *Repeatability*

Repeatability merupakan kemampuan *instrument* dalam menampilkan ulang output pengukuran yang sama pada pengukuran yang berulang.

f. Linearitas

Linearitas merupakan grafik yang menampilkan pengukuran nilai sebenarnya (input) yang dapat menghasilkan output nilai yang ditunjukkan oleh instrument (output). Pengukuran yang baik adalah ketika input pengukuran dan output pengukuran berbanding lurus (linear). Linearitas dapat diperoleh dari persamaan berikut:

$$O - O_{\min} = \frac{O_{\max} - O_{\min}}{I_{\max} - I_{\min}} (I - I_{\min}) \quad 2.3$$

Persamaan Linearitas :

$$O_{\text{ideal}} = KI + a \quad 2.4$$

Dimana:

$$K = \frac{O_{max} - O_{min}}{I_{max} - I_{min}} \quad 2.5$$

Dan

$$a = O_{min} - KI_{min} \quad 2.6$$

dimana :

$$O = 1.6 \times 10^{-3}I + 4.0 \quad 2.7$$

2.15. Teori Ketidakpastian

Ketidakpastian pengukuran merupakan tingkat seberapa besar ketidakpastian yang dihasilkan oleh suatu alat ukur. Dalam menghitung ketidakpastian pengukuran ada beberapa langkah yang harus dihitung, antara lain:

a. Koreksi

Koreksi dapat diperoleh dengan persamaan berikut.

Koreksi = Pembacaan standard-Pembacaan alat

b. Standard deviasi

$$\sigma = \frac{\sqrt{\Sigma(Di - D')^2}}{n-1} \quad 2.8$$

dimana :

Di = koreksi alat ukur

Di' = rata-rata koreksi

n = Banyak range pengukuran

c. Analisa Type A, (Ua)

Pada analisa tipe A ini hasilnya diperoleh dari data pengukuran. Adapun persamaannya adalah sebagai berikut:

$$Ua1 = \frac{\sigma_{maks}}{\sqrt{n}} \quad 2.9$$

$$Ua2 = \sqrt{\frac{SSR}{n-2}} \quad 2.10$$

d. Analisa Type B, (Ub)

Analisa tipe B ini diperoleh berdasarkan sertifikat kalibrasi atau spesifikasi dari alat ukur. Adapun persamaannya adalah sebagai berikut:

$$Ub1 = \frac{Resolusi/2}{\sqrt{3}} \quad 2.11$$

Dimana :

$SSR = \text{Sum Square Residual}$

$$Ub2 = \frac{a}{k} \quad 2.12$$

e. Ketidakpastian Kombinasi (UC)

Uc merupakan Ketidakpastian kombinasi dari ketidakpastian tipe A dan ketidakpastian tipe B. Adapun persamaan dari ketidakpastian kombinasi adalah:

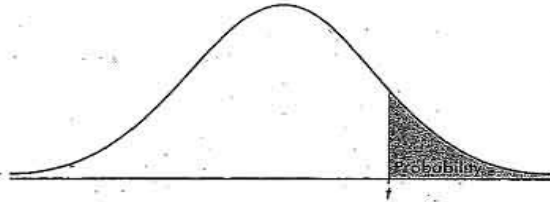
$$UC = \sqrt{Ua1^2 + Ua2^2 + Ub1^2 + Ub2^2} \quad 2.13$$

f. Ketidakpastian Diperluas

Hasil akhir kalibrasi adalah ketidakpastian diperluas sehingga alat ukur tersebut dapat diketahui ketidakpastiannya melalui U_{expand} . Persamaan U_{expand} adalah:

$$U_{expand} = k.Uc \quad 2.14$$

Untuk mencari nilai k, maka melihat table t student sesuai dengan *confidence level* 95%. Tabel T *student* dapat dilihat pada gambar 2.14.

TABLE B: *t*-DISTRIBUTION CRITICAL VALUES

| df | Tail probability <i>p</i> | | | | | | | | | | | |
|------|---------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | .25 | .20 | .15 | .10 | .05 | .025 | .02 | .01 | .005 | .0025 | .001 | .0005 |
| 1 | 1.000 | 1.376 | 1.963 | 3.078 | 6.314 | 12.71 | 15.89 | 31.82 | 63.66 | 127.3 | 318.3 | 636.6 |
| 2 | .816 | 1.061 | 1.386 | 1.886 | 2.920 | 4.303 | 4.849 | 6.965 | 9.925 | 14.09 | 22.33 | 31.60 |
| 3 | .765 | .978 | 1.250 | 1.638 | 2.353 | 3.182 | 3.482 | 4.541 | 5.841 | 7.453 | 10.21 | 12.92 |
| 4 | .741 | .941 | 1.190 | 1.533 | 2.132 | 2.776 | 2.999 | 3.747 | 4.604 | 5.598 | 7.173 | 8.610 |
| 5 | .727 | .920 | 1.156 | 1.476 | 2.015 | 2.571 | 2.757 | 3.365 | 4.032 | 4.773 | 5.893 | 6.869 |
| 6 | .718 | .906 | 1.134 | 1.440 | 1.943 | 2.447 | 2.612 | 3.143 | 3.707 | 4.317 | 5.208 | 5.959 |
| 7 | .711 | .896 | 1.119 | 1.415 | 1.895 | 2.365 | 2.517 | 2.998 | 3.499 | 4.029 | 4.785 | 5.408 |
| 8 | .706 | .889 | 1.108 | 1.397 | 1.860 | 2.306 | 2.449 | 2.896 | 3.355 | 3.833 | 4.501 | 5.041 |
| 9 | .703 | .883 | 1.100 | 1.383 | 1.833 | 2.262 | 2.398 | 2.821 | 3.250 | 3.690 | 4.297 | 4.781 |
| 10 | .700 | .879 | 1.093 | 1.372 | 1.812 | 2.228 | 2.359 | 2.764 | 3.169 | 3.581 | 4.144 | 4.587 |
| 11 | .697 | .876 | 1.088 | 1.363 | 1.796 | 2.201 | 2.328 | 2.718 | 3.106 | 3.497 | 4.025 | 4.437 |
| 12 | .695 | .873 | 1.083 | 1.356 | 1.782 | 2.179 | 2.303 | 2.681 | 3.055 | 3.428 | 3.930 | 4.318 |
| 13 | .694 | .870 | 1.079 | 1.350 | 1.771 | 2.160 | 2.282 | 2.650 | 3.012 | 3.372 | 3.852 | 4.221 |
| 14 | .692 | .868 | 1.076 | 1.345 | 1.761 | 2.145 | 2.264 | 2.624 | 2.977 | 3.326 | 3.787 | 4.140 |
| 15 | .691 | .866 | 1.074 | 1.341 | 1.753 | 2.131 | 2.249 | 2.602 | 2.947 | 3.286 | 3.733 | 4.073 |
| 16 | .690 | .865 | 1.071 | 1.337 | 1.746 | 2.120 | 2.235 | 2.583 | 2.921 | 3.252 | 3.686 | 4.015 |
| 17 | .689 | .863 | 1.069 | 1.333 | 1.740 | 2.110 | 2.224 | 2.567 | 2.898 | 3.222 | 3.646 | 3.965 |
| 18 | .688 | .862 | 1.067 | 1.330 | 1.734 | 2.101 | 2.214 | 2.552 | 2.878 | 3.197 | 3.611 | 3.922 |
| 19 | .688 | .861 | 1.066 | 1.328 | 1.729 | 2.093 | 2.205 | 2.539 | 2.861 | 3.174 | 3.579 | 3.883 |
| 20 | .687 | .860 | 1.064 | 1.325 | 1.725 | 2.086 | 2.197 | 2.528 | 2.845 | 3.153 | 3.552 | 3.850 |
| 21 | .686 | .859 | 1.063 | 1.323 | 1.721 | 2.080 | 2.189 | 2.518 | 2.831 | 3.135 | 3.527 | 3.819 |
| 22 | .686 | .858 | 1.061 | 1.321 | 1.717 | 2.074 | 2.183 | 2.508 | 2.819 | 3.119 | 3.505 | 3.792 |
| 23 | .685 | .858 | 1.060 | 1.319 | 1.714 | 2.069 | 2.177 | 2.500 | 2.807 | 3.104 | 3.485 | 3.768 |
| 24 | .685 | .857 | 1.059 | 1.318 | 1.711 | 2.064 | 2.172 | 2.492 | 2.797 | 3.091 | 3.467 | 3.745 |
| 25 | .684 | .856 | 1.058 | 1.316 | 1.708 | 2.060 | 2.167 | 2.485 | 2.787 | 3.078 | 3.450 | 3.725 |
| 26 | .684 | .856 | 1.058 | 1.315 | 1.706 | 2.056 | 2.162 | 2.479 | 2.779 | 3.067 | 3.435 | 3.707 |
| 27 | .684 | .855 | 1.057 | 1.314 | 1.703 | 2.052 | 2.158 | 2.473 | 2.771 | 3.057 | 3.421 | 3.690 |
| 28 | .683 | .855 | 1.056 | 1.313 | 1.701 | 2.048 | 2.154 | 2.467 | 2.763 | 3.047 | 3.408 | 3.674 |
| 29 | .683 | .854 | 1.055 | 1.311 | 1.699 | 2.045 | 2.150 | 2.462 | 2.756 | 3.038 | 3.396 | 3.659 |
| 30 | .683 | .854 | 1.055 | 1.310 | 1.697 | 2.042 | 2.147 | 2.457 | 2.750 | 3.030 | 3.385 | 3.646 |
| 40 | .681 | .851 | 1.050 | 1.303 | 1.684 | 2.021 | 2.123 | 2.423 | 2.704 | 2.971 | 3.307 | 3.551 |
| 50 | .679 | .849 | 1.047 | 1.299 | 1.676 | 2.009 | 2.109 | 2.403 | 2.678 | 2.937 | 3.261 | 3.496 |
| 60 | .679 | .848 | 1.045 | 1.296 | 1.671 | 2.000 | 2.099 | 2.390 | 2.660 | 2.915 | 3.232 | 3.460 |
| 80 | .678 | .846 | 1.043 | 1.292 | 1.664 | 1.990 | 2.088 | 2.374 | 2.639 | 2.887 | 3.195 | 3.416 |
| 100 | .677 | .845 | 1.042 | 1.290 | 1.660 | 1.984 | 2.081 | 2.364 | 2.626 | 2.871 | 3.174 | 3.390 |
| 1000 | .675 | .842 | 1.037 | 1.282 | 1.646 | 1.962 | 2.056 | 2.330 | 2.581 | 2.813 | 3.098 | 3.300 |
| ∞ | .674 | .841 | 1.036 | 1.282 | 1.645 | 1.960 | 2.054 | 2.326 | 2.576 | 2.807 | 3.091 | 3.291 |
| | 50% | 60% | 70% | 80% | 90% | 95% | 96% | 98% | 99% | 99.5% | 99.8% | 99.9% |
| | Confidence level <i>C</i> | | | | | | | | | | | |

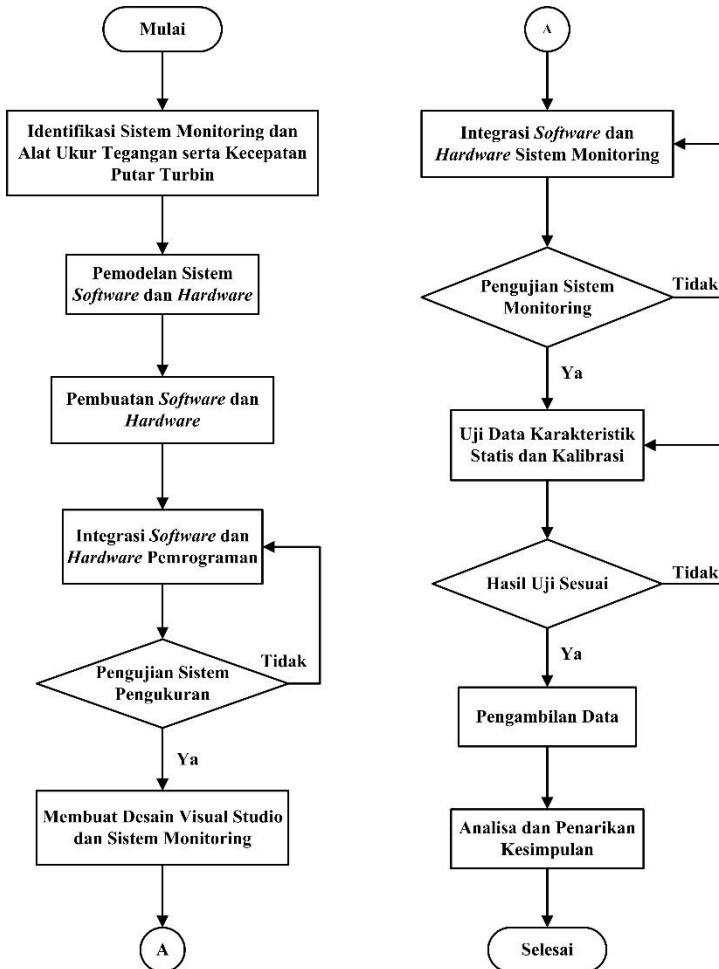
Gambar 2.14 Tabel T-student

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

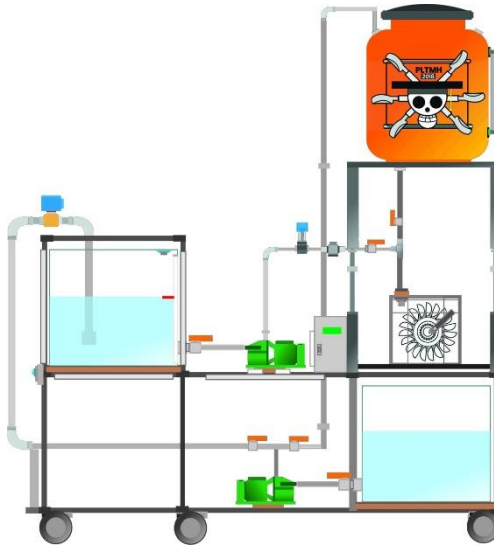
3.1. Flowchart Perancangan Alat

Langkah-langkah perancangan alat ini digambarkan dalam *flowchart* penelitian yang dapat dilihat pada gambar 3.1. di bawah ini.



Gambar 3.1 *Flowchart* Pengerjaan Tugas Akhir

Desain mini *plant* PLTMH dibuat sedemikian rupa, berikut merupakan desain mini *plant* PLTMH.

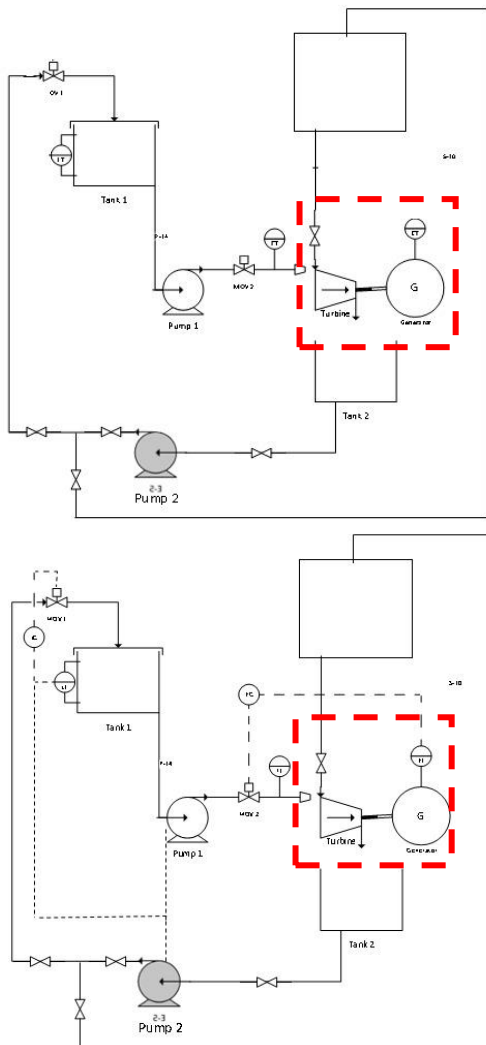


Gambar 3.2 Desain mini *plant* PLTMH

Dari gambar diatas maka dapat dibuat sebuah *piping and instrumentation diagram* pada mini *plant* PLTMH seperti pada gambar dibawah ini.

Pada *plant* ini dibuat desain *process flow diagram* seperti pada Gambar 3.3. Pada *process flow diagram* dapat dijelaskan sebuah proses alir dimana pada awalnya sensor flow akan melakukan akuisi data pada aliran air, lalu data yang didapat diteruskan ke mikrokontroler yang akan melakukan perhitungan algoritma dan memerintahkan pada aktuator untuk mengatur bukaan sesuai dengan debit yang dibutuhkan.

Setelah desain dari *Proses Flow Diagram* dibuat maka langkah selanjutnya yaitu membuat desain *Piping and Instrument Diagram* seperti pada gambar 3.3.



Gambar 3.3 PFD dan P&ID mini *plant* PLTMH

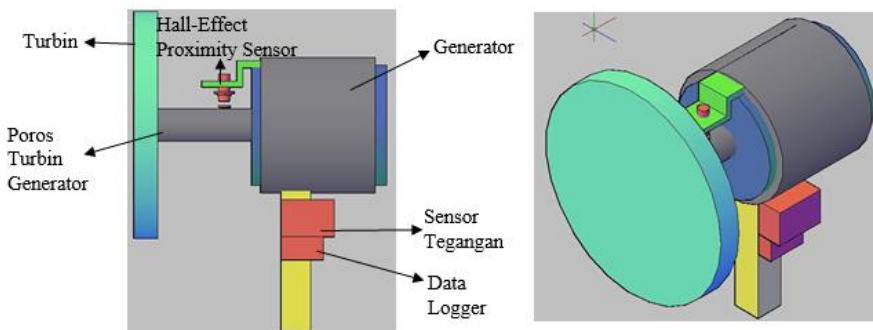
3.2 Identifikasi Sistem Monitoring dan Alat Ukur Tegangan serta RPM

Pada identifikasi meliputi mencari dan mempelajari bahan pustaka maupun konsep – konsep yang berkaitan dengan permasalahan mengenai perancangan alat sistem monitoring kecepatan turbin dan tegangan keluaran generator pada PLTMH seperti Sensor *Proximity Hall-Effect* NJK 500-2A, LCD 20×4 cm, Sensor Tegangan, Mikrokontroler ATmega 128, dan Lampu Indikator 5W12V.

3.3 Pemodelan *Hardware* dan *Software*

Pada pemodelan ini dilakukan perancangan *hardware* dan *software* terlebih dahulu untuk sistem monitoring kecepatan turbin dan tegangan keluaran generator pada PLTMH. Sebelum melakukan pembuatan alat, seperti piranti mekanik, *interface* dan *record* data hasil monitoring. Diharapkan dengan dibuatnya perancangan tersebut supaya lebih mudah dalam hal pembuatan alat nantinya.

Desain yang dibuat adalah sebagai berikut:

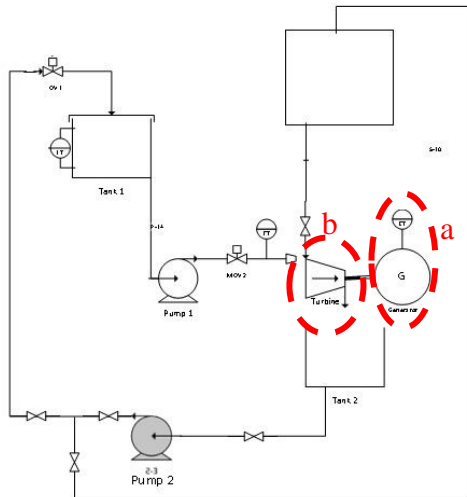


Gambar 3.4 Pemodelan Sistem Tampak Samping dan Atas

3.4 Pembuatan *Software* dan *Hardware*

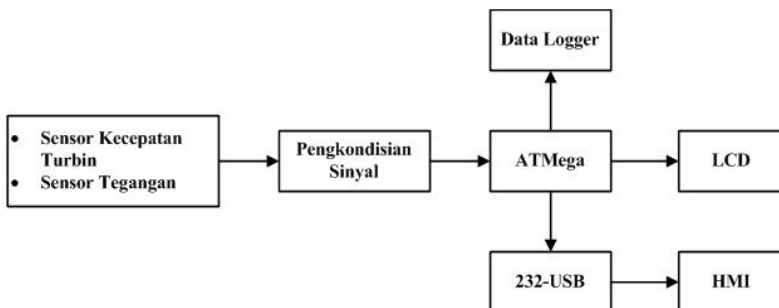
Rancang bangun Pembangkit Listrik Tenaga Mikro-Hidro terdiri dari enam bagian, yakni desain *plant system*, desain turbin, pengendalian *level*, pengendalian *flow*, sistem monitoring *level* dan

flow, dan terakhir sistem monitoring RPM dan tegangan. Desain rancang bangun PLTMH pada Gambar 3.5.



Gambar 3.5 Peletakan sensor RPM (a) dan sensor tegangan (b) pada *plant* PLTMH

Dilakukan pembuatan sistem monitoring kecepatan turbin dan tegangan keluaran generator pada PLTMH, pembuatan sistem monitoring dilakukan dengan membuat *hardware* dan *software* .



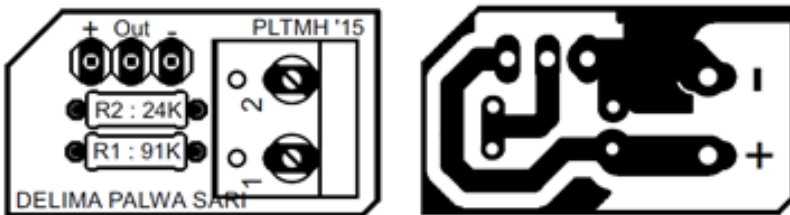
Gambar 3.6 Diagram blok sistem monitoring kecepatan turbin dan *voltage* generator pada PLTMH

Berikut ini adalah penjelasan diagram blok identifikasi dan pemodelan sistem *monitoring*.

1. Data input dari *PLTMH System Plant* berasal dari sensor dan alat ukur berupa *rpm magnetic sensor (Hall Effect Sensor)* dan sensor tegangan yang telah diolah pada mikrokontroler AVR Atmega sistem pengendalian masing-masing kemudian dikirimkan ke data logger (yang nantinya akan menyimpan data pada sdcard) dan ke komputer menggunakan kabel USB RS-232.
2. Komputer mengirimkan data input ke HMI (yang dibuat dengan menggunakan Visual Studio).
3. *Software* yang telah dibuat berfungsi sebagai display pada pc yang dapat menampilkan data, grafik, serta kabel serial yang terhubung pada pc.

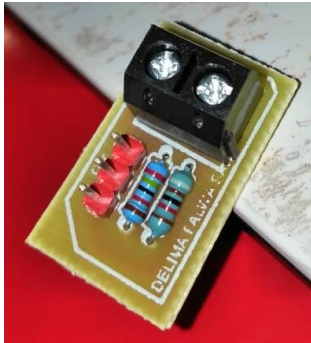
a. Perancangan dan Pembuatan Alat (*Hardware*)

Alat ukur tegangan menggunakan sensor tegangan. Dengan spesifikasi tegangan inputannya maksimum 100 V. Dikarenakan tegangan inputan maksimumnya yakni 100 V jadi menggunakan rangkaian pembagi tegangan, untuk menghindari kerusakan pada atmega. Penambahan kapasitor untuk menstabilkan tegangan keluaran yang masuk pada mikrokontroler tidak diperlukan dikarenakan tegangan inputannya DC. Skematik rangkaian modul sensor tegangan dapat dilihat pada Gambar 3.7.



Gambar 3.7 Skematik Rangkaian Modul Sensor Tegangan pada *Software EAGLE*

Skematik diatas kemudian di print pada *board* PCB dan hasilnya pada gambar 3.8.



Gambar 3.8 Modul Sensor Tegangan



Gambar 3.9 Mini *Plant* PLTMH

b. Perancangan dan Pembuatan Alat (*Software*)

Perancangan *software* pada tahap ini merupakan siklus gabungan untuk rangkaian pemrosesan sinyal. Rangkaian pemrosesan sinyal pada alat ini menggunakan ATMega128. Untuk membuat program pada ATMega128, dibutuhkan *software CodeVisionAVR*. Pada program ini, akan diberikan koding untuk mengolah sinyal masukan.

Dengan langkah – langkah sebagai berikut.

1. Alat dan bahan (Hall-Effect Proximity Sensor NJK 500-2A, Sensor Tegangan, Downloader USB ISP, Kabel Jumper, Mikrokontroler ATMega 128, LCD 20 × 4 cm dan modul OpenLog) disiapkan.
2. Laptop yang akan dipakai, pastikan sudah terinstall *software CV AVR*.

Untuk pemrograman menggunakan *software Code Vision AVR* dilakukan dengan langkah-langkah sebagai berikut:

- a. *Software Code Vision AVR* dibuka.



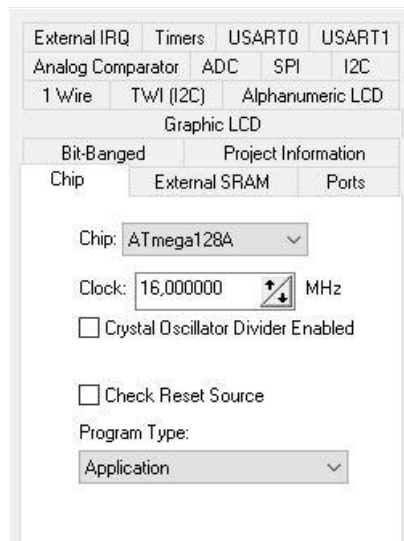
Gambar 3.10 Buka *Software*

- b. *File* dipilih, kemudian *New* dipilih. Kemudian akan muncul gambar seperti dibawah ini.



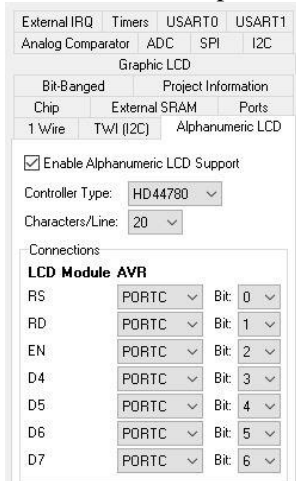
Gambar 3.11 *Create New File*

- c. Pada *File Type*, *Project* dipilih, kemudian OK dipilih.
- d. Pada *Chip*, ATmega128 dipilih dan pada frekuensi *clock* diisi seperti gambar 3.13.



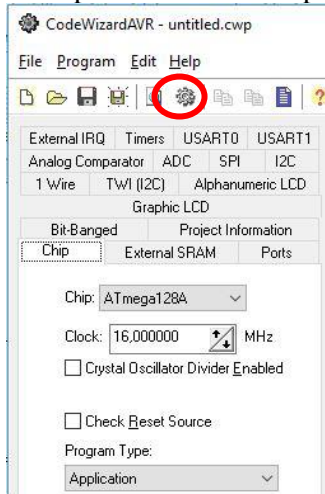
Gambar 3.12 *Chip pada Code Vision AVR*

- e. Pada *Alphanumeric LCD*, *Enable Alphanumeric LCD Support* dipilih, kemudian pada *Characters/Line* dipilih 20, dan pada *LCD Module AVR* dipilih *PORTC*.



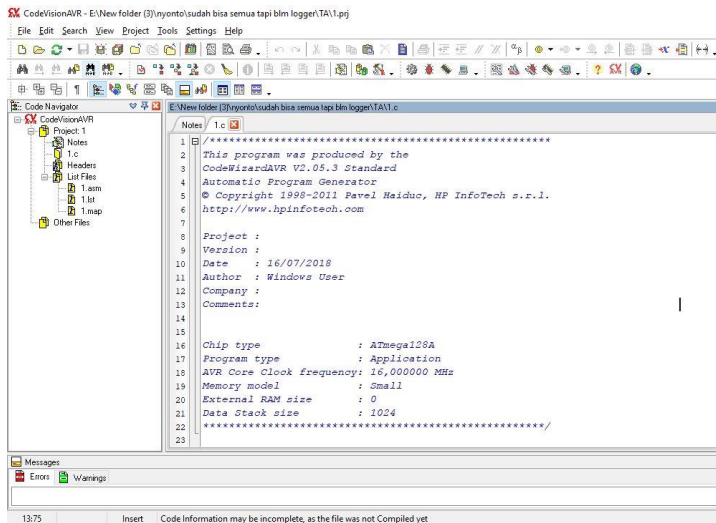
Gambar 3.13 *Alphanumeric LCD* pada *Code Vision AVR*

- f. Kemudian tombol seperti dibawah ini dipilih.



Gambar 3.14 Tombol *Generate*

- g. Setelah itu akan muncul program yang telah diatur dan pemrograman bisa dilakukan.



Gambar 3.15 Tampilan Awal Program

- h. Setelah selesai menyusun program, program dapat di-*compile*.
3. Atmega128 dihubungkan dengan kabel downloader ditancapkan pada isp port atmega untuk disambungkan pada laptop.
 4. Datasheet ATmega 128, dicek terlebih dahulu.
 5. Sensor tegangan disiapkan dan dihubungkan pada ATmega128.

Sensor tegangan dipasang secara paralel terhadap sumber tegangan. Module sensor tegangan menggunakan prinsip kerja pembagi tegangan. Memiliki range pembacaan 0 - 100 VDC dan output 0 – 5 VDC, Kemudian pin analog pada modul disambungkan ke kontroler pada pin PF.0. Untuk mendapatkan

hasil pembacaan tegangan, mikrokontroler harus diprogram terlebih dahulu agar data analog dari sensor dapat dibaca dan berubah menjadi digital.

| | |
|------------------------------------|--|
| <code>void baca_volt(){</code> | : sebuah deklarasi fungsi <code>baca_volt</code> yang dapat dipanggil untuk mewakili listing program didalamnya. |
| <code>for(i=0;i<10;i++){</code> | : yaitu sebuah fungsi perulangan untuk mengulang sebanyak 10 kali listing program yang ada didalamnya. |
| <code>adc=read_adc(0);</code> | : membaca data analog pada pin PF.0 ATmega128. |
| <code>av_adc=av_adc+adc; }</code> | : sebuah formulasi penjumlahan data yaitu <code>av_adc</code> setelah pengambilan data sebanyak 10 kali. |
| <code>av_adc=av_adc/10;</code> | : sebuah formulasi rata rata data <code>av_adc</code> yang sudah diambil sebanyak 10 kali. |
| <code>volt=av_adc*100/255;</code> | : formulasi untuk perhitungan nilai tegangan yang masuk pada sensor yang kemudian akan ditampilkan. |

Dari program diatas dapat diketahui bahwa untuk mendapatkan hasil pembacaan Tegangan digunakan rumus Tegangan = $ADC \times 100 / 255$. Dimana tegangan ADC adalah pembacaan sinyal analog dari sensor tegangan dikali 100 yang merupakan nilai maksimum dari besaran fisis yang dapat *disensing* oleh sensor dan 255 merupakan jumlah dari bit mikrokontroller ATmega128 yang digunakan.

6. Pin pada Sensor *Hall Effect Proximity* NJK 500-2A dihubungkan ke atmega dengan menggunakan kabel jumper.
Sensor rpm yang digunakan adalah Sensor *Hall Effect Proximity* NJK 500-2A sensor ini dipasang dengan range maksimal

1 cm. Kemudian pin digital disambungkan ke controller yaitu pin PE.4 agar dapat menghitung kecepatan putar turbin dari sensor, mikrokontroler harus diprogram terlebih dahulu seperti dibawah ini.

```
interrupt [EXT_INT4] void : mengindikasikan bahwa
ext_int4_isr(void)        listing program yang ada
                           didalamnya berada pada
                           pin interrupt 4 pada
                           ATmega128 dan akan
                           bekerja setiap pin interrupt
                           4 mengalami perubahan.
```

```
counter++;                : sebuah variable counter agar
                           counter untuk listing
                           program didalamnya terus
                           bertambah seiring dengan
                           perubahan nilai tegangan
                           pada pin interrupt 4
                           ATmega128 dari rendah ke
                           tinggi (karena menggunakan
                           opsi rising edge) .
```

```
interrupt [TIM0_OVF] void : mengindikasikan bahwa
timer0_ovf_isr(void){     listing program yang ada
                           didalamnya berjalan sesuai
                           dengan timer yang telah
                           diatur sebelumnya.
```

```
waktu++;                  : sebuah variable bernama
                           waktu agar waktu untuk
                           listing program
                           didalamnya terus
                           bertambah seiring dengan
                           berjalannya timer.
```

```
if (waktu >=120){         : merupakan pengkondisian
                           apabila waktu  $\geq$  120 maka
                           waktu sama dengan 1
                           detik.
```

| | |
|------------------------------------|--|
| <code>Rpm = counter * 60.0;</code> | : formulasi perhitungan untuk rpm dikalikan 60 untuk merubah satuan detik menjadi menit. |
| <code>Rpm2 = Rpm * 4.0;</code> | : formulasi perhitungan untuk rpm dikalikan 4 karena turbin menggunakan gear box dengan perbandingan gear 1 : 4. |
| <code>counter = 0;</code> | : agar nilai variable counter pada saat program di running akan selalu dikembalikan menjadi 0, setiap waktu ≥ 120 . |
| <code>waktu = 0;</code> | : agar nilai variable waktu pada saat program di running akan dikembalikan menjadi 0, setiap waktu ≥ 120 . |
| <code>Rpm2 = Rpm2;}}</code> | : agar nilai rpm tidak ikut ter-reset. |

7. Kemudian pin modul OpenLog modul dihubungkan ke pin ATMega 128.

SD card berfungsi sebagai penyimpan data logger pembacaan kecepatan putar turbin dan tegangan serta disimpan dalam format .txt. Konfigurasi pin yang dihubungkan ke pin ATMega 128 adalah Tx, Rx, GND, Vcc dengan pin pada ATMega PD.2, PD.3, GND, 5V. SD CARD menggunakan format FAT32 dan memiliki kapasitas memori 2GB.

| | |
|--|---|
| <code>sprintf(buff3, "%02u:%02u:%02u/%02u:%02u:%02u ; %.2f cm ; %d RPM ; %.2f lps ; %.2f V\r",dd,mm,yy,s,m,h, jrk, Rpm, flow_rate, volt);</code> | : untuk menampung nilai nilai (dd:mm:yy/h:m:s;level;RPM; flow;volt) kedalam variable buff3 yang bertipe data char |
|--|---|

`poutput = USART1;` : mengindikasikan bahwa USART1 akan digunakan sebagai media komunikasi serial penyimpanan data pada SD Card melalui modul `openlog`.

`puts(buff3);` : untuk mengirim data yang ada pada variable `buff3` ke modul `openlog` melalui komunikasi serial.

8. Pin modul RTC modul dihubungkan ke pin ATmega 128.

Komponen Realtime clock adalah komponen IC penghitung yang dapat difungsikan sebagai sumber data waktu baik berupa data jam, hari, bulan maupun tahun. Bentuk komunikasi data dari IC RTC adalah I2C yang merupakan kepanjangan dari Inter Integrated Circuit. Komunikasi jenis ini hanya menggunakan 2 jalur komunikasi yaitu SCL dan SDA. Konfigurasi pin yang dihubungkan ke pin ATmega 128 adalah GND dihubungkan ke GND, 5V dihubungkan ke 5V, SDA dihubungkan ke PD.0, dan SCL dihubungkan ke PD.1. Untuk mendapatkan hasil pembacaan RTC, mikrokontroler harus diprogram terlebih dahulu.

`h=22;m=28;s=00;` : berfungsi mengatur waktu awal pembacaan RTC dan digunakan sebagai titik mulai melakukan perhitungan waktu.

`rtc_set_time(h,m,s);`

`dd=16;mm=07;yy=18;` : berfungsi mengatur tanggal awal pembacaan RTC dan digunakan sebagai titik mulai melakukan

`rtc_set_date(wd,dd,mm,yy);`

| | |
|--|---|
| | perhitungan tanggal. |
| <code>rtc_get_time(&h,&m,&s);</code> | : merupakan kode untuk mendapatkan waktu pembacaan RTC. |
| <code>rtc_get_date(&wd,&dd,&mm,&yy);</code> | : merupakan kode untuk mendapatkan tanggal pembacaan RTC. |
| <code>sprintf(buff3, "%02u:%02u:%02u/%02u:%02u:%02u\n",dd,mm,yy,s,m,h);</code> | : memindahkan nilai yang ada pada variable-variable tanggal dan waktu ke dalam variabel buff3 yang bertipe data char. |
| <code>poutput = USART1;</code> | : mengindikasikan bahwa USART1 akan digunakan sebagai media pengiriman data listng program dibawahnya |
| <code>puts(buff3);</code> | : menampilkan data yang tertampung pada variabel buff3 dengan komunikasi serial. |

9. Pin LCD 20×4 cm modul dihubungkan ke pin ATmega 128, dengan ketentuan :
 - VSS dihubungkan ke GND
 - V0 dihubungkan ke keluaran Potensio
 - VDD dihubungkan ke 5V
 - Rs dihubungkan ke PC.0

- Rw dihubungkan ke PC.1
- E dihubungkan ke PC.2
- D4 dihubungkan ke PD.3
- D5 dihubungkan ke PC.4
- D6 dihubungkan ke PC.5
- D7 dihubungkan ke PC.6

10. Kodingan disiapkan dan diupload pada atmega 128 dengan menggunakan *software* Khazama.

Kemudian Kabel sensor dan pin modul OpenLog dipasang sesuai dengan penempatan

3.5 Integrasi *Software* dan *Hardware* Pemrograman

Ketika koding program sudah jadi maka dilakukan pengintegrasian antara *hardware* dengan *software*. Didalam *software* khazama dilakukan penyamaan mikrokontroler yang digunakan untuk melakukan proses *uploading* program yang ada ke ATMega128.

Penyesuaian *interface* antara *software* CV AVR dan *hardware* ATMega128 dapat dilihat pada LCD ataupun dan komunikasi serial yang ditampilkan pada *software* yang telah dibuat pada visual studio serta dilakukan pengecekan data yang tersimpan pada SD Card.

3.6 Pengujian Sistem Pengukuran

Pengujian sistem pengukuran ini dilakukan untuk mengetahui sudah berjalankah atau belum hasil *uploading* program dari *software* CV AVR ke *hardware* ATMega128. Masing – masing dari *sensing element* dicoba untuk mengukur. Jika dari masing – masing *sensing element* masih belum dapat menampilkan data, maka proses pemrograman dan integrasi *software* serta *hardware* perlu diulang.

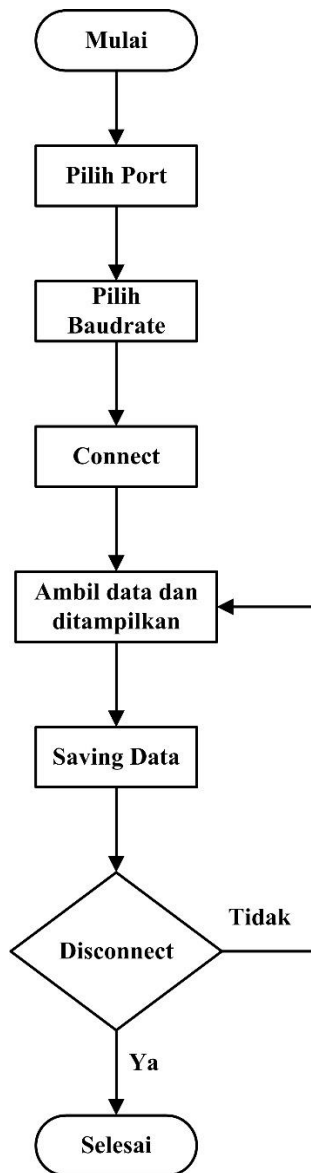
3.7 Pembuatan Desain HMI pada Visual Basic dan Data Logger

Pada poin ini terdapat beberapa poin terkait perancangan software yaitu sebagai berikut:

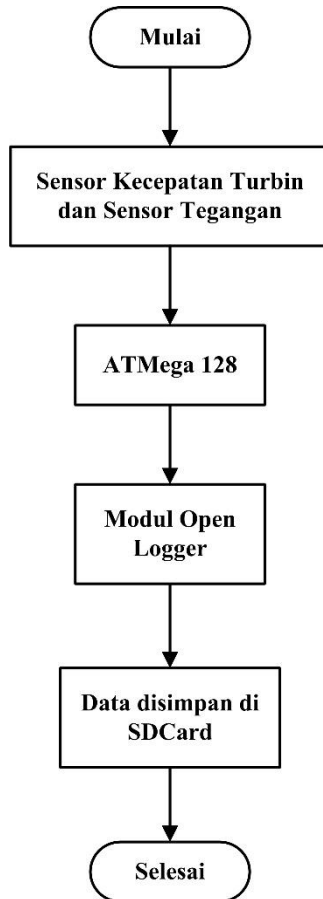
a. HMI

HMI atau Human Machine Interface berfungsi sebagai perantara antara User dan Mesin disini HMI dibuat berupa aplikasi monitoring yang dapat menerima dan membaca data kecepatan turbin dan tegangan dari sistem PLTMH serta menampilkannya pada PC, data yang akan ditampilkan secara *real time*. Untuk pembuatan aplikasi dilakukan pada Software Visual Basic menggunakan Bahasa pemrograman basic. Berikut adalah flowchart pembuatan HMI.

Dari flowchart di atas dapat diketahui bahwa proses yang ada pada HMI ini yaitu dimulai dengan memilih port yang sudah terhubung dengan laptop, setelah port dipilih lalu memilih baudrate yang akan dipakai, baudrate yang biasanya dipakai adalah 9600 bps. Setelah memilih baudrate maka klik tombol connect, setelah connect maka data kecepatan putar turbin dan tegangan akan muncul secara realtime sesuai dengan kondisi yang ada di plant. Pada HMI yang saya buat juga terdapat fasilitas database yang dapat menyimpan data data tersebut langsung menyimpan ke PC saat kita sudah mengklik tombol save. Jika penggunaan HMI selesai maka klik tombol disconnect dan koneksi antara plant dan PC akan terputus.



Gambar 3.16 Flowchart HMI

b. Data Logger

Gambar 3.17 Flowchart proses penyimpanan data pada Open Logger

SD card berfungsi sebagai penyimpan data logger menggunakan modul open logger, pembacaan kecepatan putar turbin dan tegangan disimpan dalam format .txt. SD CARD menggunakan format FAT32 dan memiliki kapasitas memori 2GB. Berikut adalah flowchart proses penyimpanan data.

Dari *flowchart* di bawah dapat diketahui bahwa proses penyimpanan data pada OpenLogger ini yaitu dimulai dengan data dari sensor rpm dan tegangan ditransmisikan ke ATmega 128 untuk diproses dan dikondisikan sinyalnya. Setelah itu data tersebut akan dikirim ke modul OpenLogger dengan melalui komunikasi serial USART 1. Data yang dikirim serial ke modul tersebut akan diproses dan nantinya data tersebut akan disimpan di SD Card dalam bentuk file .txt.

3.8 Integrasi Hardware, Software dan Sistem Monitoring

Integrasi *hardware* dan *software* pada *visual studio*, dan sistem penyimpanan *datalogger* pada SD Card.

3.9 Pengambilan Data Karakteristik Statik dan Kalibrasi

Dalam tahap ini merupakan melihat spesifikasi yang dimiliki oleh sistem monitoring yang telah dibuat. Dimana dengan keterangan data tersebut dapat dilihat bagus atau tidaknya performansi karakter sistem monitoring ini. Pada karakteristik statik alat yang dicari yaitu nilai *range*, *span*, *resolusi*, *sensitivitas*, *linieritas*, *hysteresis*, serta prosentase akurasi. Sedangkan untuk data kalibrasi digunakan untuk mencari nilai ketidakpastian dari hasil pengukuran ketika menggunakan perangkat ini.

3.10 Analisa Data dan Penarikan Kesimpulan

Tahap terakhir terdapat analisis data yang dilakukan bertujuan untuk mengetahui kinerja dari setiap komponen apakah sesuai dengan perancangan *hardware* dan *software* dari hasil pengukuran, apabila tidak sesuai maka akan di cek ulang pada uji karakteristik statis dan kalibrasi. Kemudian, dilakukan penarikan kesimpulan dari semua elemen yang mempengaruhi data tersebut.

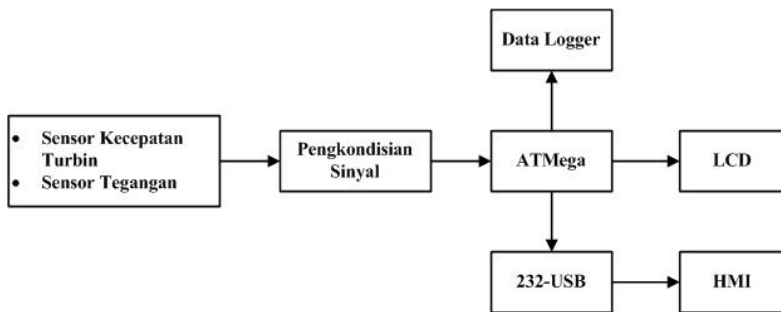
Halaman ini sengaja dikosongkan

BAB IV

PENGUJIAN ALAT DAN ANALISIS DATA

4.1 Hasil Pengujian Alat

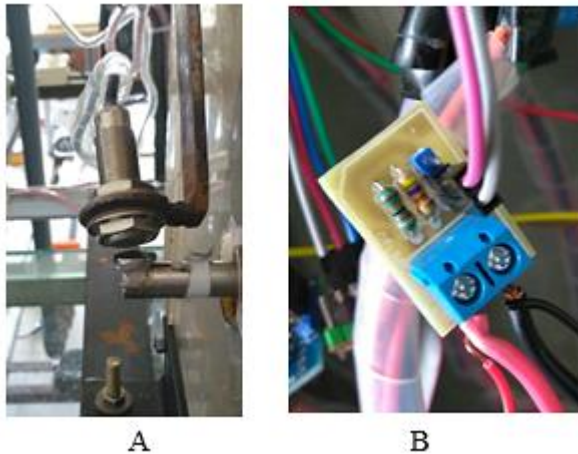
Sistem monitoring adalah sistem untuk mengamati hasil perubahan suatu besaran fisis pada objek yang ingin kita ukur, serta dapat diinformasikan berupa tampilan atau biasa disebut HMI (Human Machine Interfacing). Sehingga pengguna dapat mengamati perubahan besaran fisis dengan mudah secara *real time*. Berikut ini adalah diagram blok monitoring kecepatan turbin dan *voltage* generator pada PLTMH yang ditunjukkan oleh gambar 4.1.



Gambar 4.1 Diagram blok sistem monitoring kecepatan turbin dan *voltage* generator pada PLTMH

Berdasarkan Gambar 4.1 terdapat berbagai komponen monitoring pada mini plant PLTMH untuk memonitoring nilai monitoring kecepatan turbin dan *voltage* generator yang dihasilkan oleh PLTMH, nilai ini harus akurat dan real time agar data yang didapat dapat menjadi acuan apabila ingin diolah lebih lanjut dan dikirim melalui komunikasi data serial. Telah direalisasikan *hardware* sesuai dengan diagram blok di atas untuk mengetahui apakah sistem monitoring kecepatan turbin dan *voltage* generator pada PLTMH yang dirancang telah memenuhi harapan atau belum.

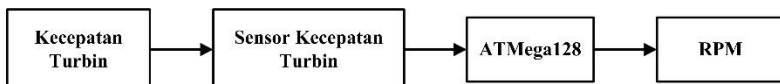
4.1.1. Realisasi Pemasangan Sensor



Gambar 4.2 Realisasi Pemasangan Sensor RPM (A) dan Sensor Tegangan (B)

Dari gambar di atas dapat diketahui bahwa sensor yang dipakai untuk mengukur kecepatan putar turbin yaitu sensor *Proximity Hall Effect NJK-5002A*. Sensor tersebut dipasang pada poros turbin generator dengan diberi pelindung berupa *seal* plastik agar tidak terjadi kerusakan akibat grounding pada sensor dan pada kabel diberi spiral agar rapi dan aman. Sedangkan sensor tegangan dipasang pada kabel keluaran generator.

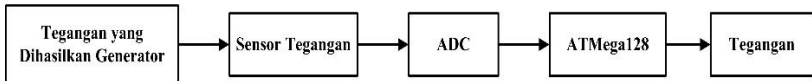
4.1.2. Pemrosesan Data



Gambar 4.3 Data Processing Sensor Kecepatan Putar Turbin

Dari gambar di atas dapat diketahui besaran fisis yang dideteksi dan akan disensing oleh sensor Proximity Hall Effect

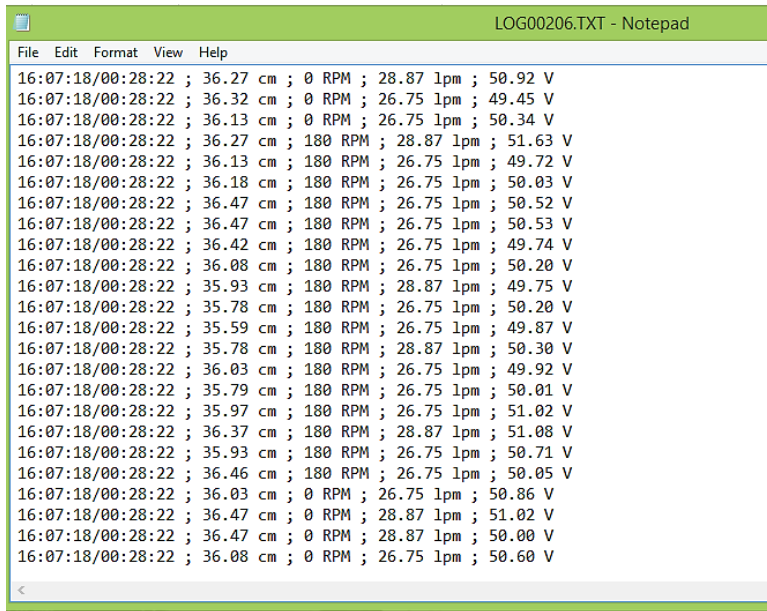
adalah putaran turbin dengan menggunakan prinsip seperti halnya *switch*, saat sensor mengenai magnet maka sensor akan mengirimkan data digital berupa tegangan, jika tidak mengenai magnet maka sensor tidak akan *men-sensing* data apapun. Jadi setiap ada magnet sensor akan terus *meng-counter* sebanyak jumlah putaran atau sebanyak sensor mengenai magnet tersebut. Data output dari sensor ini adalah data digital berupa tegangan yang nantinya data tersebut dapat secara langsung ditransmisikan pada ATmega 128 untuk diproses, lalu akan muncul data berupa angka dengan satuan ‘rpm’ .



Gambar 4.4 *Data Processing Sensor Tegangan*

Dari gambar di atas dapat diketahui besaran fisis yang dideteksi dan akan disensing oleh sensor tegangan adalah tegangan yang dihasilkan generator dengan menggunakan prinsip seperti saat sistem sedang *running* sensor akan mendeteksi tegangan kemudian tegangan tersebut akan diturunkan oleh rangkaian pembagi tegangan pada modul sensor. Penurunan tegangan tersebut akan sebanding dengan V_{input} . Data output dari sensor ini adalah data analog berupa tegangan yang nantinya data tersebut akan diproses terlebih dahulu pada rangkaian ADC mikrokontroler, agar dapat ditransmisikan pada ATmega 128 untuk diproses, lalu akan muncul data berupa angka dengan satuan ‘Volt’ .

4.1.3. Data Logger



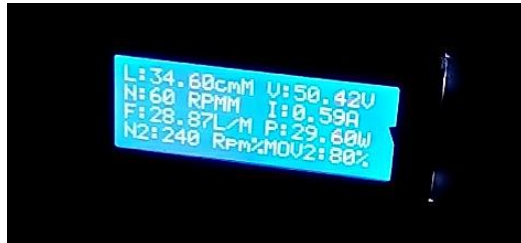
Gambar 4.5 Tampilan hasil penyimpanan Data Logger menggunakan Modul OpenLog

Dari gambar di atas dapat diketahui ada 5 macam data yang ditampilkan dengan pembatas “;” yaitu :

- Pertama dengan format h:m:s/dd:mm:yy : merupakan waktu dan tanggal pengambilan data.
- Kedua dengan satuan cm : merupakan data pembacaan sensor level.
- Ketiga dengan satuan rpm : merupakan data pembacaan sensor kecepatan putar turbin.
- Keempat dengan satuan lpm : merupakan data pembacaan sensor flow.
- Kelima dengan satuan V : merupakan data pembacaan sensor tegangan.

Data yang diambil dari pembacaan keseluruhan sensor di atas disimpan dalam SD Card dengan format “.txt”

4.1.4. Tampilan LCD



Gambar 4.6 Tampilan LCD pada Panel

Dari gambar di atas dapat diketahui bahwa tampilan LCD sistem monitoring untuk mini plant PLTMH memiliki beberapa data yang ditampilkan seperti.

- L : menampilkan data hasil pembacaan level.
- N : menampilkan data hasil pembacaan kecepatan putar turbin.
- F : menampilkan data hasil pembacaan *flow*.
- N2 : menampilkan untuk data hasil pembacaan kecepatan putar generator.
- V : menampilkan data hasil pembacaan tegangan yang dihasilkan generator
- I : menampilkan data hasil pembacaan arus.
- P : menampilkan untuk data hasil pembacaan daya.
- MOV : menampilkan data hasil pembacaan bukaan MOV.

4.1.5. HMI (Human Machine Interfacing)



Gambar 4.7 Tampilan menu utama HMI

Dari gambar di atas dapat diketahui bahwa HMI sistem monitoring untuk mini plant PLTMH memiliki beberapa tab seperti.

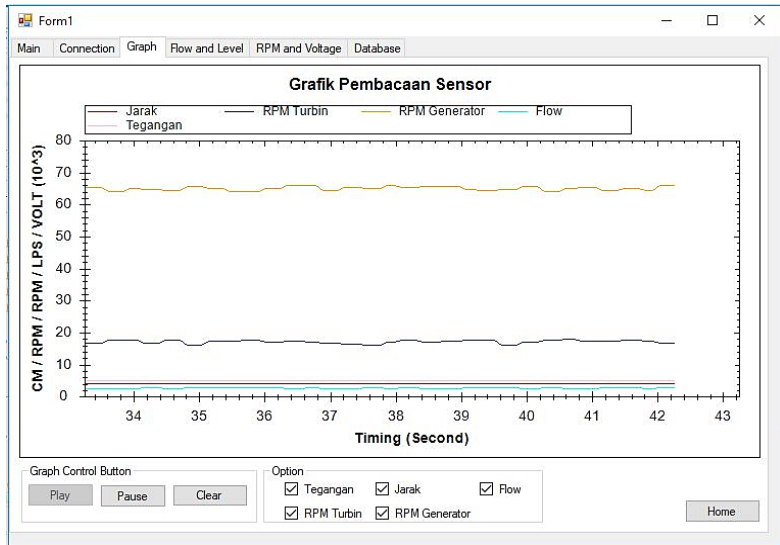
- Connection : tab yang berisi pengaturan koneksi dan memuat gambar plan keseluruhan.
- Grafik : tab untuk menuju laman yang menampilkan grafik.
- Flow dan level : tab yang berisi tentang gambar plan sensor flow dan level beserta data pembacaan sensor yang telah diolah dan ditampilkan secara real time.
- RPM dan voltage : yaitu tab yang memuat gambar plan sensor kecepatan putaran turbin dan tegangan beserta data pembacaan sensor yang telah diolah dan ditampilkan secara real time.
- Database : yaitu tab yang memuat data keseluruhan sensor berupa list table, yang nantinya akan disimpan kedalam PC.



Gambar 4.8 Tampilan untuk tab Connection

Dari gambar di atas dapat diketahui bahwa HMI mini plant PLTMH pada tab Connection terdapat beberapa fitur seperti.

- **COM** : pada combo box ini berfungsi untuk memilih port hardware yang terhubung dengan PC.
- **Baudrate** : pada combo box ini berfungsi untuk memilih baudrate yang akan digunakan dengan perangkat yang nantinya akan dipasang.
- **Tombol Connect** : berfungsi untuk memulai koneksi antara hardware yang telah dipilih dengan PC.
- **Tombol Disconnect** : berfungsi untuk memutus koneksi antara hardware yang sudah terhubung dengan PC.
- **Tombol Home** : berfungsi untuk kembali ke tab utama atau main menu.



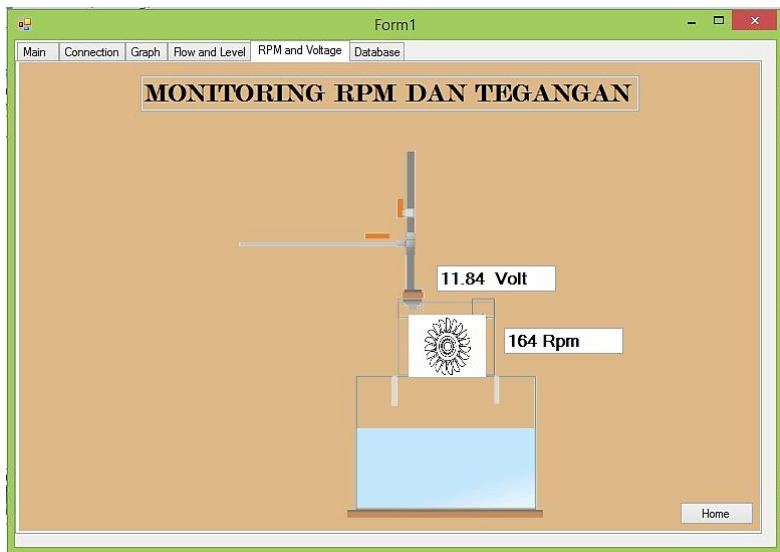
Gambar 4.9 Tampilan untuk tab Grafik

Dari gambar di atas dapat diketahui bahwa HMI mini plant PLTMH pada tab Grafik terdapat beberapa fitur seperti.

- Check box Tegangan : apabila check box ini dicentang, maka grafik data pembacaan sensor tegangan akan muncul dan bergerak.
- Check box RPM Turbin : apabila check box ini dicentang, maka grafik data pembacaan sensor kecepatan putaran turbin akan muncul dan bergerak.
- Check box Jarak : apabila check box ini dicentang, maka grafik data pembacaan sensor level akan muncul dan bergerak.
- Check box RPM Generator : apabila check box ini dicentang, maka grafik data pembacaan sensor kecepatan putaran generator akan muncul dan bergerak.
- Check box Flow : apabila check box ini dicentang, maka grafik data pembacaan sensor flow akan muncul dan bergerak.
- Tombol Play : merupakan tombol yang berfungsi untuk memulai mengerjakan grafik, sehingga grafik yang check box

nya sudah dicentang dapat bergerak sesuai dengan data pembacaan sensornya.

- Tombol Pause : merupakan tombol yang berguna untuk menghentikan laju pergerakan grafik tanpa harus memutus sambungan dengan hardware yang terhubung dengan PC.
- Tombol Clear : merupakan tombol yang akan membersihkan layar grafik dari grafik data pembacaan sensor.
- Tombol Home : berfungsi untuk kembali ke tab utama atau main menu.

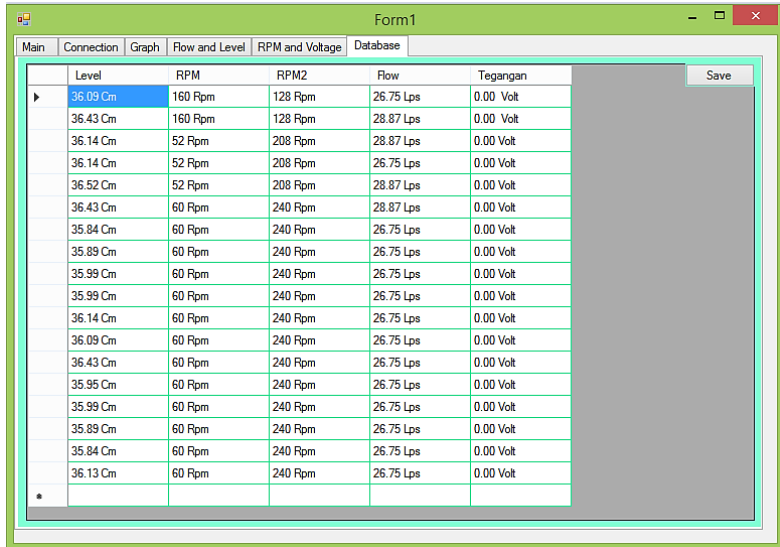


Gambar 4.10 Tampilan untuk tab Monitoring Kecepatan Turbin dan Tegangan

Dari gambar di atas dapat diketahui bahwa HMI mini plant PLTMH pada tab Monitoring Kecepatan Turbin dan Tegangan terdapat beberapa fitur seperti.

- Textbox 1 : berisi data pembacaan sensor tegangan yang sudah diolah dan data yang ditampilkan bersifat real time sesuai dengan keadaan plan.

- Textbox 2 : berisi data pembacaan sensor kecepatan turbin yang sudah diolah dan data yang ditampilkan bersifat real time sesuai dengan keadaan plan.
- Tombol Home : berfungsi untuk kembali ke tab utama atau main menu.



| Level | RPM | RPM2 | Flow | Tegangan | Save |
|----------|---------|---------|-----------|-----------|------|
| 36.09 Cm | 160 Rpm | 128 Rpm | 26.75 Lps | 0.00 Volt | |
| 36.43 Cm | 160 Rpm | 128 Rpm | 28.87 Lps | 0.00 Volt | |
| 36.14 Cm | 52 Rpm | 208 Rpm | 28.87 Lps | 0.00 Volt | |
| 36.14 Cm | 52 Rpm | 208 Rpm | 26.75 Lps | 0.00 Volt | |
| 36.52 Cm | 52 Rpm | 208 Rpm | 28.87 Lps | 0.00 Volt | |
| 36.43 Cm | 60 Rpm | 240 Rpm | 28.87 Lps | 0.00 Volt | |
| 35.84 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |
| 35.89 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |
| 35.99 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |
| 35.99 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |
| 36.14 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |
| 36.09 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |
| 36.43 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |
| 35.95 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |
| 35.99 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |
| 35.89 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |
| 35.84 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |
| 36.13 Cm | 60 Rpm | 240 Rpm | 26.75 Lps | 0.00 Volt | |

Gambar 4.11 Tampilan untuk tab Database

Dari gambar di atas dapat diketahui bahwa HMI mini plant PLTMH pada tab Database terdapat beberapa fitur seperti.

- Kolom Level : berisi pembacaan data sensor level yang direkam dari awal sistem terkoneksi dengan hardware.
- Kolom RPM : berisi pembacaan data sensor kecepatan putaran turbin yang direkam dari awal sistem terkoneksi dengan hardware.
- Kolom RPM2 : berisi pembacaan data sensor kecepatan putaran generator yang direkam dari awal sistem terkoneksi dengan hardware.
- Kolom Flow : berisi pembacaan data sensor flow yang direkam dari awal sistem terkoneksi dengan hardware.

- Kolom Tegangan : berisi pembacaan data sensor tegangan yang direkam dari awal sistem terkoneksi dengan hardware.
- Tombol Save : berfungsi menyimpan data pembacaan sensor pada tabel kedalam PC dengan format “.xls”.

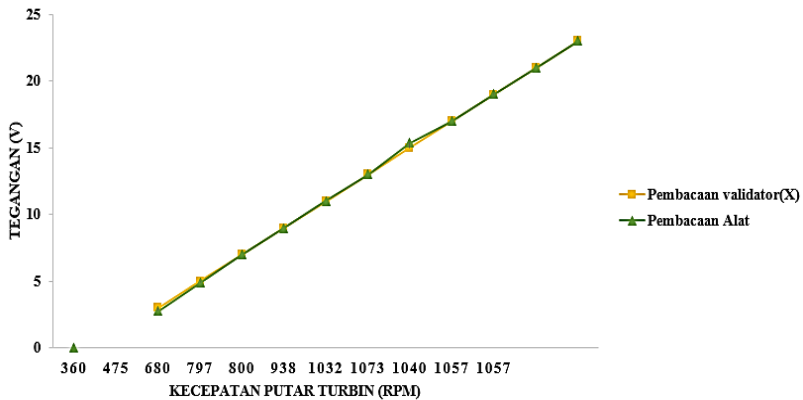
4.2 Hasil Uji Karakteristik Alat

4.2.1 Karakteristik Stasik Sensor Tegangan

Pengujian sensor tegangan dilakukan pada rentang tegangan 3V – 23V dengan media beberapa alat elektronik. Pada setiap kenaikan, diambil data sebanyak 5 kali pembacaan sensor. Alat Standar yang digunakan untuk pembanding adalah AVO meter standar. Berikut ini data yang diperoleh dari pengujian alat.

Tabel 4.1 Data Sensor Tegangan

| Pembacaan Validator (V) | Pembacaan Alat | | Rata-rata Pembacaan (V) | Koreksi (V) |
|-------------------------|----------------|-----------|-------------------------|-------------|
| | Naik (V) | Turun (V) | | |
| 3 | 2.788 | 2.774 | 2.781 | 0.226 |
| 5 | 4.878 | 5.038 | 4.985 | -0.038 |
| 7 | 6.998 | 7.056 | 7.027 | -0.056 |
| 9 | 9.01 | 9.024 | 9.017 | -0.024 |
| 11 | 11.022 | 11.068 | 11.045 | -0.068 |
| 13 | 13.016 | 12.994 | 13.005 | 0.006 |
| 15 | 15.354 | 15.122 | 15.238 | -0.122 |
| 17 | 16.986 | 17.022 | 17.004 | - 0.022 |
| 19 | 19.036 | 19.03 | 19.033 | - 0.03 |
| 21 | 20.992 | 21.04 | 21.016 | - 0.04 |
| 23 | 23.016 | 23.102 | 23.06 | -0.102 |
| Jumlah | | | | -0.168 |
| Rata-rata | | | | -0.0168 |



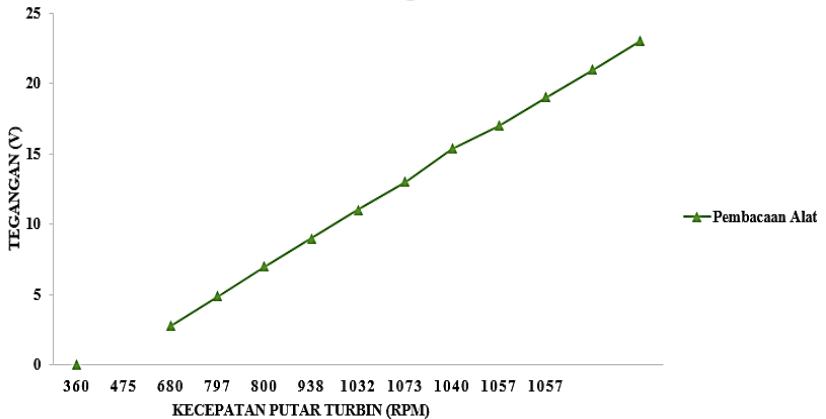
Gambar 4.12 Grafik Pembacaan Sensor Tegangan

Dari Gambar 4.8 menunjukkan bahwa pembacaan alat sudah mendekati dengan alat standar. Namun, terdapat beberapa titik yang masih terdapat *error* pembacaan.

Berikut ini nilai karakteristik statik sensor Tegangan dengan mengacu pada rumus 2.1 hingga 2.14 :

- Range : 3V – 23V
- Span : 20V
- Resolusi : 0,01
- Sensitivitas : 1.0164
- Non – Linieritas* : 33.3%.

Sehingga didapatkan non linieritas sebesar 33.3%. Didalam fungsi garis yang tidak linier ini menunjukkan perbedaan antara hasil pembacaan *actual* atau nyata dengan garis yang sesuai pada setiap kenaikan pengukuran.



Gambar 4.13 Grafik Linieritas Sensor Tegangan

f. Akurasi : **99.81 %**

g. Error : 0.19 %

4.2.2 Kalibrasi Sensor Tegangan

Berdasarkan data yang telah didapatkan dari pengujian alat ukur tegangan pada table 4.1 dengan mengacu pada rumus 2.1 hingga 2.14, berikut merupakan perhitungan untuk mencari nilai ketidakpastian alat ukur.

- Standar Deviasi (σ) = 0.087142183
- Ketidakpastian Tipe A (U_{a1}) = 0.026274357
- Nilai Ketidakpastian Regresi (U_{a2}) = 0.081352426
 $b = -0.005578698$
 $a = 0.050144$
 $SSR = 0.059563956$
- Nilai ketidakpastian tipe B
 Ketidakpastian Resolusi (U_{B1}) = 0,003
 Dikarenakan pada alat standar terdapat sertifikat kalibrasinya maka nilai a (ketidakpastian sertifikat kalibrasi) dianggap mendekati 0, dan nilai faktor cakupan dianggap 2,0.

Sehingga hasil : Ketidakpastian alat standar tegangan (U_{B2}) = 0

- Nilai ketidakpastian kombinasi $U_c = 0.085538836$
- V_{eff} (Nilai derajat kebebasan efektif) = 0.3
 $V_{eff} = 0.3$, sehingga jika dibulatkan menjadi 4, dimana pada table *T-student* menghasilkan nilai k (faktor koreksi) sebesar 6.314.
- U_{exp} (Nilai ketidakpastian diperluas) = 0.54

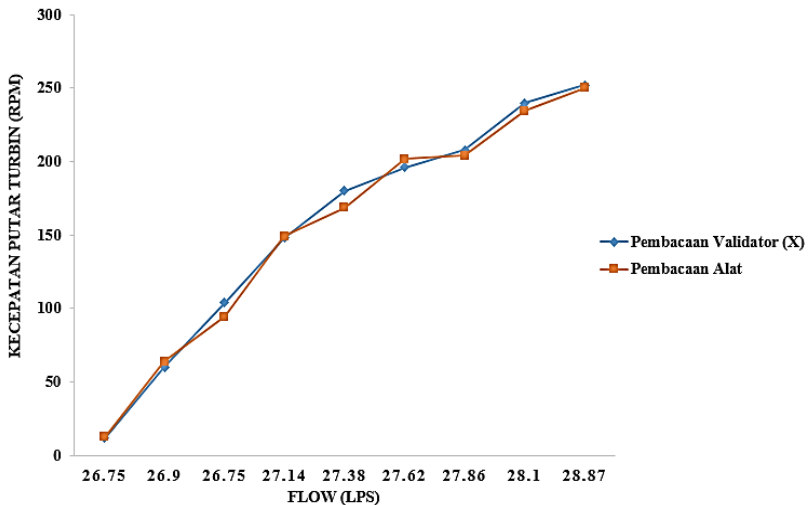
Sehingga berdasarkan perhitungan ketidakpastian diperluas diatas, menghasilkan nilai ketidakpastian alat sebesar ± 0.54 dengan tingkat kepercayaan 95% dari tabel *T-Student*. Nilai ketidakpastian tersebut akan menjadi acuan untuk pembacaan alat ukur selama alat ukur tersebut digunakan.

4.2.3 Karakteristik Statik Sensor Proximity Hall Effect NJK-5002A

Karakteristik statik sensor Proximity Hall Effect NJK-5002A didapatkan dari perbandingan nilai *output* sensor dengan Tachometer standar. Pengujian RPM dilakukan dengan pengambilan data setiap kenaikan, diambil data sebanyak 5 kali pembacaan alat dan masing masing percobaan dilakukan dalam rentan waktu 1 menit. Alat Standar yang digunakan untuk pembandingan adalah Tachometer standar. Berikut ini data yang diperoleh dari pengujian alat.

Tabel 4.2 Data Pengujian Sensor Proximity Hall Effect

| Pembacaan Validator (rpm) | Rata-rata Pembacaan | | Rata-rata Pembacaan (rpm) | Koreksi (rpm) |
|---------------------------|---------------------|-------------|---------------------------|---------------|
| | Naik (rpm) | Turun (rpm) | | |
| 12 | 12.48 | 12.68 | 12.58 | -0.68 |
| 60 | 63.8 | 67.3 | 65.55 | -7.3 |
| 104 | 94.46 | 94.46 | 94.46 | 9.54 |
| 148 | 149.36 | 149.46 | 149.41 | -1.46 |
| 180 | 168.8 | 168.8 | 168.8 | 11.2 |
| 196 | 201.6 | 201.8 | 201.7 | -5.8 |
| 208 | 204 | 202.8 | 203.4 | 5.2 |
| 240 | 234.4 | 234.4 | 234.4 | 5.6 |
| 252 | 250 | 246.4 | 248.2 | 5.6 |
| Jumlah | | | | 21.9 |
| Rata-rata | | | | 2.433333 |

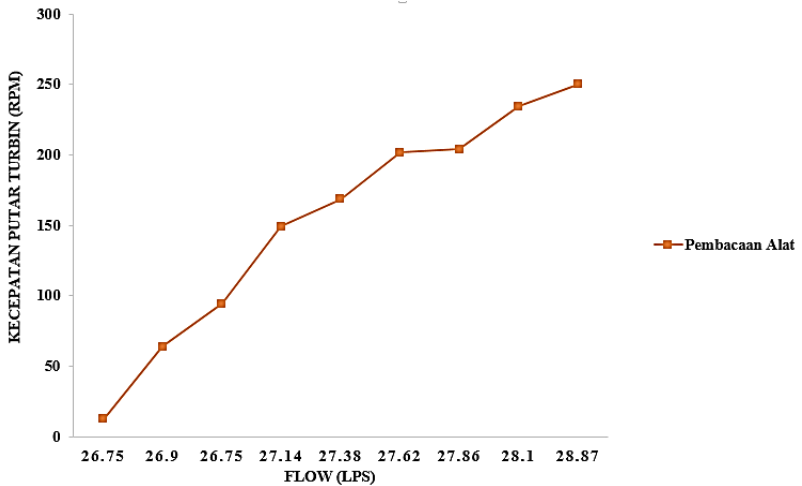


Gambar 4.14 Grafik Pembacaan Sensor Proximity Hall
Effect NJK-5002A

Berikut ini hasil perhitungan nilai karakteristik statik alat ukur RPM berdasarkan data pada tabel 4.2 dengan mengacu pada rumus 2.1 hingga 2.14 .

- Range : 12 rpm – 252 rpm
- Span : 240 rpm
- Resolusi : 0.01
- Sensitivitas : **0,98175**

Sehingga didapatkan non linieritas sebesar 7.49%. Didalam fungsi garis yang tidak linier ini menunjukkan perbedaan antara hasil pembacaan *actual* atau nyata dengan garis seperti di bawah ini.



Gambar 4.15 Grafik Linieritas Sensor Proximity Hall Effect NJK-200A

- e. Akurasi : **95,39%**
 f. Error : 4.61 %

4.2.4 Kalibrasi Sensor Proximity Hall Effect NJK-5002A

Berikut merupakan perhitungan ketidakpastian alat ukur berdasarkan tabel 4.2 dengan mengacu pada rumus 2.1 hingga 2.14.

- Standar Deviasi = 5.851885166
- Nilai ketidakpastian tipe A = 1.764409765
- Nilai ketidakpastian regresi $U_{a2} = 6.488204529$
 $b = 0.024165396$
 $a = -1.32573$
 $SSR = 294.6775861$

- Nilai ketidakpastian tipe B
Ketidakpastian Resolusi (U_{B1}) = 0,003
Dikarenakan pada alat standar terdapat sertifikat kalibrasinya maka nilai a (ketidakpastian sertifikat kalibrasi) dianggap mendekati 0, dan nilai faktor cakupan dianggap 2,0. Sehingga hasil : Ketidakpastian alat standar termometer (U_{B2}) = 0
- Nilai ketidakpastian kombinasi $U_c = 6.723834335$
- V_{eff} (Nilai derajat kebebasan efektif) = 9.177
Sehingga jika dibulatkan menjadi 9, dimana pada table *T-student* menghasilkan nilai k (faktor koreksi) sebesar 1.833.
- U_{exp} (Nilai ketidakpastian diperluas) = 12.325

Sehingga berdasarkan perhitungan ketidakpastian diperluas diatas, menghasilkan nilai ketidakpastian alat sebesar ± 12.325 dengan tingkat kepercayaan 95% dari tabel *T-Student*. Nilai ketidakpastian tersebut akan menjadi acuan untuk pembacaan alat ukur selama alat ukur tersebut digunakan.

4.3 Pembahasan

Dari data yang didapat, diketahui bahwa software HMI dapat menerima data secara real time dan dapat memisahkan data waktu, tanggal, level, kecepatan turbin, flow dan tegangan. Dengan data yang dibuat berubah-ubah berdampak pada bentuk grafik yang naik dan turun mengikuti nilai data.

Data yang kedua didapatkan dari SD Card yang berfungsi sebagai data logger yang bertujuan untuk pengarsipan data dari pengukuran sensor. Dari hasil uji coba alat didapatkan range pembacaan Tegangan sebesar 0 hingga 53.68 Volt dan kecepatan turbin sebesar 0 hingga 252 rpm. Sedangkan pada SD card didapatkan data range pembacaan Tegangan sebesar 6.57 hingga 53.68 Volt dan kecepatan turbin sebesar 12 rpm hingga 252 rpm. Terdapat perbedaan pembacaan karena pada HMI pembacaan data dimulai dan dihentikan sesuai kehendak pengguna, berbeda dengan pembacaan pada SD Card yaitu ketika sistem berjalan maka data

hasil pembacaan sensor secara otomatis akan tersimpan sebagai *logger* atau riwayat hingga sistem mati atau SD Card di lepas. Kemudian untuk uji karakteristik sensor, data pembacaan sensor tegangan diambil dengan memvariasi bacaan sensor dan membandingkannya dengan AVOMeter, didapatkan hasil bahwa modul sensor yang dibuat memiliki tingkat keakurasian sebesar 99.81 %, begitu juga dengan sensor kecepatan putar turbin data diambil dengan cara memvariasi debit air yang memutar turbin, saat debit air divariasi maka kecepatan turbinpun juga akan mengalami variasi sesuai dengan debit air yang mendorong turbin bergerak dan data hasil pembacaan sensor dibandingkan dengan pembacaan Tachometer. Didapatkan hasil bahwa sensor *Proximity Hall Effect* yang digunakan memiliki tingkat keakurasian sebesar 95.39% . Sebenarnya tingkat keakurasian sensor *Proximity Hall Effect* yaitu sebesar 99.5% , dikarenakan pergantian generator yang menyebabkan pergerakan poros turbin semakin melemah dan bacaan sensor jadi berkurang keakurasiannya.

Halaman ini sengaja dikosongkan

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari pengujian sistem monitoring kecepatan turbin dan tegangan keluaran generator pada *plant* PLTMH dapat disimpulkan, yaitu :

- a. Telah berhasil dirancang sistem monitoring kecepatan putar turbin dan tegangan keluaran generator menggunakan sensor *Proximity Hall Effect NJK500-2A* dan sensor Tegangan, yang dapat menyimpan data sebagai logger pada SD Card dengan format file.txt serta dapat menampilkan data secara *realtime* pada HMI yang sudah dibuat menggunakan *software* visual studio.
- b. Telah diuji karakteristik statik sensor, dimana sensor Tegangan memiliki nilai ketidakpastian diperluas (U_{exp}) sebesar ± 0.54 dengan tingkat kepercayaan 95%. Akurasi sensor Tegangan sebesar 99.81%, dengan *error* pembacaan sebesar 0.19%, sensitivitas sebesar 98,5%, non linieritas sebesar 10.186%. Sensor *Proximity Hall Effect NJK-5002A* memiliki nilai ketidakpastian diperluas (U_{exp}) sebesar ± 12.325 dengan tingkat kepercayaan 95%. Akurasi Sensor *Proximity Hall Effect NJK-5002A* sebesar 95.39% dengan sensitivitas sebesar 97,56%, non linieritas sebesar 7.49%.

5.2 Saran

Adapun saran untuk tugas akhir sistem monitoring kecepatan turbin dan tegangan keluaran generator pada *plant* PLTMH ini adalah sebagai berikut :

- a. Perlu dilakukan pengembangan lagi, agar sistem HMI dapat untuk mengontrol dan membaca aktuator pada plant secara *realtime*.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Wirabudi, W. I. (2016). *PENGEMBANGAN SISTEM KONTROL GATE PADA FLOW CONTROL VALVE SISTEM PEMBANGKIT DAYA MIKROHIDRO BERBASIS PLC SIEMENS S7-1200*. Surabaya: ITS.
- [2] WingWijayanto. (2012). *RANCANG BANGUN SISTEM MONITORING KECEPATAN TURBIN*, 1.
- [3] Punchstein, A.F., Liloyd,T.C., Conrad, A.G., 1960. "Alternating Current Machines". New York : John Willey and Sons Inc.
- [4] Prastowo, Indra. dkk. 2015. *HUMAN MACHINE INTERFACE (HMI) BERBASIS ANDROID UNTUK MONITORING DAN KENDALI SISTEM CATU DAYA*, Pearson Education - United States America.
- [5] Anonim. 2016. *Modul Praktikum Sistem Pengukuran Massa*. Surabaya : Departemen Teknik Instrumentasi, FV-ITS
- [6] Nurcahyo. 2012. *Aplikasi dan Teknik Pemrograman Mikrokontroler AVR Atmel CV Andi Offset*.
- [7] Dr. Sri Purnomo Sari, S. M. (n.d.). *PENGARUH JARAK DAN UKURAN NOZZLE*.
- [8] Texas Instrument. 2014. *Hall-Effect Sensor with PNP or NPN Output Manual Book*.
- [9] Hartono, Rudi.2013. *PERANCANGAN SISTEM DATA LOGGER TEMPERATUR BATERAI BERBASIS ARDUINO DUEMILANOVE*. Jember : Universitas Jember

- [10] Politeknik Negeri Elektronika Surabaya.
*bima.lecturer.pens.ac.id/materi%20praktikum%20dsp/ps2_c
odec_sampling.pdf*. n.d. bima.lecturer.pens.ac.id (accessed
January 29, 2018).

LAMPIRAN A

(DATA SHEET SYSTEM ATMEGA128)

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 133 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
 - Fully Static Operation
 - Up to 16MIPS Throughput at 16MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 128Kbytes of In-System Self-programmable Flash program memory
 - 4Kbytes EEPROM
 - 4Kbytes Internal SRAM
 - Write/Erase cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Up to 64Kbytes Optional External Memory Space
 - Programming Lock for Software Security
 - SPI Interface for In-System Programming
- QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix acquisition
 - Up to 64 sense channels
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Two 8-bit PWM Channels
 - 6 PWM Channels with Programmable Resolution from 2 to 16 Bits
 - Output Compare Modulator
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Dual Programmable Serial USARTs
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
 - Software Selectable Clock Frequency
 - ATmega103 Compatibility Mode Selected by a Fuse
 - Global Pull-up Disable
- I/O and Packages
 - 53 Programmable I/O Lines
 - 64-lead TQFP and 64-pad QFN/MLF
- Operating Voltages
 - 2.7 - 5.5V ATmega128L
 - 4.5 - 5.5V ATmega128
- Speed Grades
 - 0 - 8MHz ATmega128L
 - 0 - 16MHz ATmega128



**8-bit Atmel
Microcontroller
with 128KBytes
In-System
Programmable
Flash**

**ATmega128
ATmega128L**

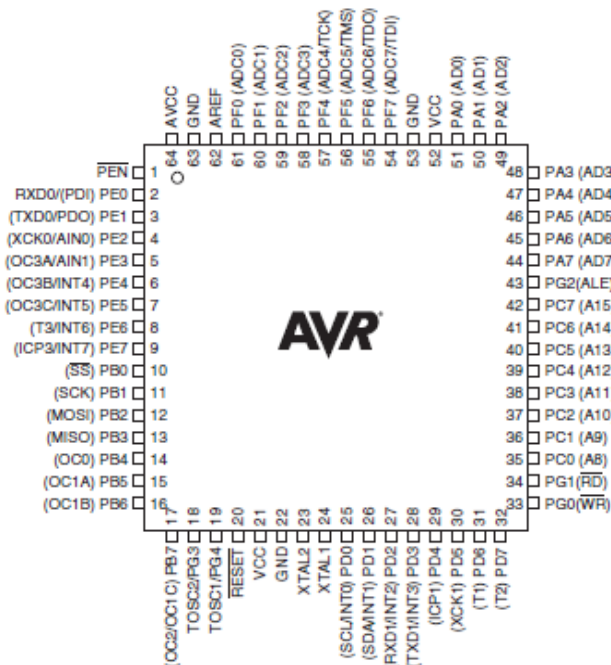
Summary

Rev. 246725-AVR-00/11



Pin Configurations

Figure 1. Pinout ATmega128



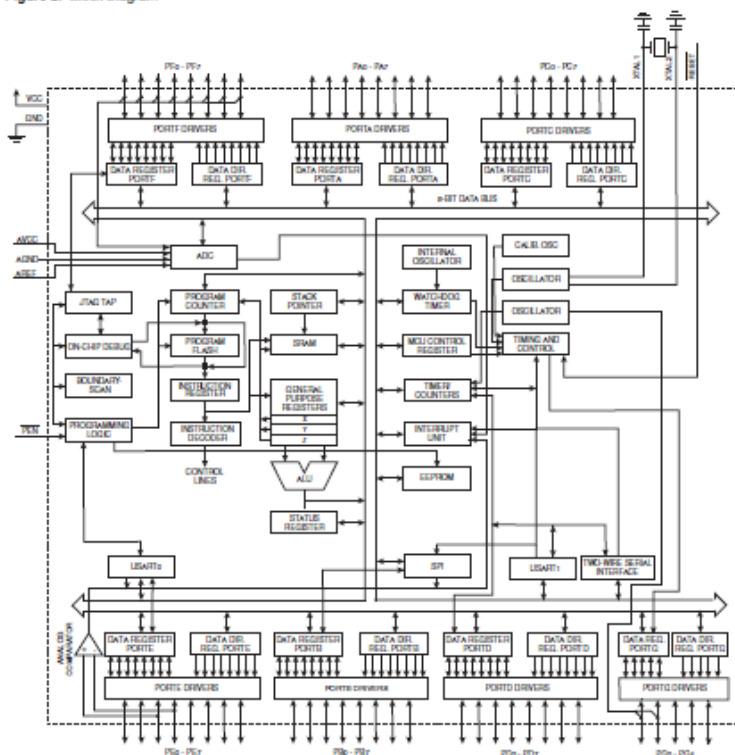
Note: The Pinout figure applies to both TQFP and MLF packages. The bottom pad under the QFN/MLF package should be soldered to ground.

Overview

The Atmel® AVR® ATmega128 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega128 achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



The Atmel® AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega128 provides the following features: 128Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 4Kbytes EEPROM, 4Kbytes SRAM, 53 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), four flexible Timer/Counters with compare modes and PWM, 2 USARTs, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega128 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega128 device is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

ATmega103 and ATmega128 Compatibility

The ATmega128 is a highly complex microcontroller where the number of I/O locations supercedes the 64 I/O locations reserved in the AVR instruction set. To ensure backward compatibility with the ATmega103, all I/O locations present in ATmega103 have the same location in ATmega128. Most additional I/O locations are added in an Extended I/O space starting from \$60 to \$FF, (i.e., in the ATmega103 internal RAM space). These locations can be reached by using LD/ST/STDD and ST/ST/STD instructions only, not by using IN and OUT instructions. The relocation of the internal RAM space may still be a problem for ATmega103 users. Also, the increased number of interrupt vectors might be a problem if the code uses absolute addresses. To solve these problems, an ATmega103 compatibility mode can be selected by programming the fuse M103C. In this mode, none of the functions in the Extended I/O space are in use, so the internal RAM is located as in ATmega103. Also, the Extended Interrupt vectors are removed.

ATmega103 Compatibility Mode

The ATmega128 is 100% pin compatible with ATmega103, and can replace the ATmega103 on current Printed Circuit Boards. The application note "Replacing ATmega103 by ATmega128" describes what the user should be aware of replacing the ATmega103 by an ATmega128.

By programming the M103C fuse, the Atmel® ATmega128 will be compatible with the ATmega103 regards to RAM, I/O pins and interrupt vectors as described above. However, some new features in ATmega128 are not available in this compatibility mode, these features are listed below:

- One USART instead of two, Asynchronous mode only. Only the eight least significant bits of the Baud Rate Register is available.
- One 16 bits Timer/Counter with two compare registers instead of two 16-bit Timer/Counters with three compare registers.
- Two-wire serial interface is not supported.
- Port C is output only.
- Port G serves alternate functions only (not a general I/O port).
- Port F serves as digital input only in addition to analog input to the ADC.
- Boot Loader capabilities is not supported.
- It is not possible to adjust the frequency of the internal calibrated RC Oscillator.
- The External Memory Interface can not release any Address pins for general I/O, neither configure different wait-states to different External Memory Address sections.

In addition, there are some other minor differences to make it more compatible to ATmega103:

- Only EXTRF and PORF exists in MCUCSR.
- Timed sequence not required for Watchdog Time-out change.
- External Interrupt pins 3 - 0 serve as level interrupt only.
- USART has no FIFO buffer, so data overrun comes earlier.

Unused I/O bits in ATmega103 should be written to 0 to ensure same operation in ATmega128.

Pin Descriptions

VCC Digital supply voltage.

GND Ground.

Port A (PA7..PA0) Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATmega128 as listed on [page 72](#).

Port B (PB7..PB0) Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega128 as listed on [page 73](#).

Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C also serves the functions of special features of the Atmel® AVR® ATmega128 as listed on [page 76](#). In ATmega103 compatibility mode, Port C is output only, and the port C pins are not tri-stated when a reset condition becomes active.

Note: The ATmega128 is by default shipped in ATmega103 compatibility mode. Thus, if the parts are not programmed before they are put on the PCB, PORTC will be output during first power up, and until the ATmega103 compatibility mode is disabled.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega128 as listed on [page 77](#).

Port E (PE7..PE0)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E also serves the functions of various special features of the ATmega128 as listed on [page 80](#).

Port F (PF7..PF0)

Port F serves as the analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a Reset occurs.

The TDO pin is tri-stated unless TAP states that shift out data are entered.

Port F also serves the functions of the JTAG interface.

In ATmega103 compatibility mode, Port F is an input Port only.

Port G (PG4..PG0)

Port G is a 5-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also serves the functions of various special features.

The port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

In ATmega103 compatibility mode, these pins only serves as strobes signals to the external memory as well as input to the 32KHz Oscillator, and the pins are initialized to PG0 = 1, PG1 = 1, and PG2 = 0 asynchronously when a reset condition becomes active, even if the clock is not running. PG3 and PG4 are oscillator pins.

| | |
|--------------|--|
| RESET | Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 19 on page 50 . Shorter pulses are not guaranteed to generate a reset. |
| XTAL1 | Input to the inverting Oscillator amplifier and input to the internal clock operating circuit. |
| XTAL2 | Output from the inverting Oscillator amplifier. |
| AVCC | AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. |
| AREF | AREF is the analog reference pin for the A/D Converter. |
| PEN | PEN is a programming enable pin for the SPI Serial Programming mode, and is internally pulled high. By holding this pin low during a Power-on Reset, the device will enter the SPI Serial Programming mode. PEN has no function during normal operation. |

Resources

A comprehensive set of development tools, application notes, and datasheets are available for download on <http://www.atmel.com/avr>.

Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C

About Code Examples

This datasheet contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

For I/O registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBR", "SBR", "SBR", and "CBR".

Capacitive touch sensing

The Atmel QTouch Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR microcontrollers. The QTouch Library includes support for the QTouch and QMatrix acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

The QTouch Library is FREE and downloadable from the Atmel website at the following location: www.atmel.com/qtouchlibrary. For implementation details and other information, refer to the [Atmel QTouch Library User Guide](#) - also available for download from the Atmel website.

Instruction Set Summary

| Mnemonic | Operands | Description | Operation | Flags | #Cycles |
|--|----------|--|-----------------------------------|---------------|---------|
| ARITHMETIC AND LOGIC INSTRUCTIONS | | | | | |
| ADD | rd, r | Add two Registers | $Res = Rd + Rr$ | Z, C, N, V, H | 1 |
| ADC | rd, r | Add with Carry two Registers | $Res = Rd + Rr + C$ | Z, C, N, V, H | 1 |
| ADW | rd, k | Add Immediate to Word | $IndirectRd = IndirectRd + K$ | Z, C, N, V, S | 2 |
| SUB | rd, r | Subtract two Registers | $Res = Rd - Rr$ | Z, C, N, V, H | 1 |
| SUBS | rd, k | Subtract Constant from Register | $Res = Rd - K$ | Z, C, N, V, H | 1 |
| SBC | rd, r | Subtract with Carry two Registers | $Res = Rd - Rr - C$ | Z, C, N, V, H | 1 |
| SBCS | rd, k | Subtract with Carry Constant from Reg | $Res = Rd - K - C$ | Z, C, N, V, H | 1 |
| SDW | rd, k | Subtract Immediate from Word | $IndirectRd = IndirectRd - K$ | Z, C, N, V, S | 2 |
| AND | rd, r | Logical AND Registers | $Res = Rd \& Rr$ | Z, N, V | 1 |
| ANDI | rd, k | Logical AND Register and Constant | $Res = Rd \& K$ | Z, N, V | 1 |
| ORI | rd, r | Logical OR Registers | $Res = Rd \vee Rr$ | Z, N, V | 1 |
| ORI | rd, k | Logical OR Register and Constant | $Res = Rd \vee K$ | Z, N, V | 1 |
| LCR | rd, r | Exclusive OR Registers | $Res = Rd \oplus Rr$ | Z, N, V | 1 |
| COM | rd | One's Complement | $Res = \sim Rd$ | Z, C, N, V | 1 |
| NEG | rd | Two's Complement | $Res = \sim Rd + 1$ | Z, C, N, V, H | 1 |
| SEDI | rd, k | Set Sign in Register | $Res = Rd \vee K$ | Z, N, V | 1 |
| CLR | rd, k | Clear Sign in Register | $Res = Rd \& \sim K$ | Z, N, V | 1 |
| INC | rd | Increment | $Res = Rd + 1$ | Z, N, V | 1 |
| DEC | rd | Decrement | $Res = Rd - 1$ | Z, N, V | 1 |
| TEST | rd | Test for Zero or Minus | $Res = Rd \& Rd$ | Z, N, V | 1 |
| CLP | rd | Clear Register | $Res = Rd \& \sim Rd$ | Z, N, V | 1 |
| SRP | rd | Set Register | $Res = Rd \vee Rd$ | None | 1 |
| MUL | rd, r | Multiply Unsigned | $Result = Rd \times Rr$ | Z, C | 2 |
| MULS | rd, r | Multiply Signed | $Result = Rd \times Rr$ | Z, C | 2 |
| MULSU | rd, r | Multiply Signed with Unsigned | $Result = Rd \times Rr$ | Z, C | 2 |
| FIMUL | rd, r | Fractional Multiply Unsigned | $Result = (Rd \times Rr) \ll 1$ | Z, C | 2 |
| FIMULS | rd, r | Fractional Multiply Signed | $Result = (Rd \times Rr) \ll 1$ | Z, C | 2 |
| FIMULSU | rd, r | Fractional Multiply Signed with Unsigned | $Result = (Rd \times Rr) \ll 1$ | Z, C | 2 |
| BRANCH INSTRUCTIONS | | | | | |
| JMP | k | Relative Jump | $PC = PC + k + 1$ | None | 2 |
| LJMP | k | Indirect Jump to ZP | $PC = Z$ | None | 2 |
| JMP | k | Direct Jump | $PC = k$ | None | 3 |
| ICALL | k | Relative Subroutine Call | $PC = PC + k + 1$ | None | 3 |
| ICALL | k | Indirect Call to ZP | $PC = Z$ | None | 3 |
| LCALL | k | Direct Subroutine Call | $PC = k$ | None | 4 |
| RET | | Subroutine Return | $PC = STACK$ | None | 4 |
| RETI | | Interrupt Return | $PC = STACK$ | 1 | 4 |
| CPSE | rd, r | Compare, Skip if Equal | $if (Rd == Rr) PC = PC + 2 or 3$ | None | 1/2/3 |
| CP | rd, r | Compare | $Res = Rr$ | Z, N, V, C, H | 1 |
| CPD | rd, r | Compare with Carry | $Res = Rr - C$ | Z, N, V, C, H | 1 |
| CPDI | rd, k | Compare Register with Immediate | $Res = K$ | Z, N, V, C, H | 1 |
| SEDF | rd, r | Skip if Set in Register Cleared | $if (Rd \& Rr) PC = PC + 2 or 3$ | None | 1/2/3 |
| SEDF | rd, r | Skip if Set in Register is Set | $if (Rd \& Rr) PC = PC + 2 or 3$ | None | 1/2/3 |
| SEDI | r, k | Skip if Set in Register Cleared | $if (Rr \& K) PC = PC + 2 or 3$ | None | 1/2/3 |
| SEDI | r, k | Skip if Set in Register is Set | $if (Rr \& K) PC = PC + 2 or 3$ | None | 1/2/3 |
| SEDF | k, k | Branch if Status Flag Set | $if (SRFL \& K) PC = PC + 2 or 3$ | None | 1/2 |
| SEDF | k, k | Branch if Status Flag Cleared | $if (SRFL \& K) PC = PC + 2 or 3$ | None | 1/2 |
| SEDI | k | Branch if Equal | $if (Z == 0) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Not Equal | $if (Z == 1) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Carry Set | $if (C == 1) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Carry Cleared | $if (C == 0) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Set or Higher | $if (Rr > Rd) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Lower | $if (Rr < Rd) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Minus | $if (N == 1) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Plus | $if (N == 0) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Greater or Equal, Signed | $if (Rr \ge Rd) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Less Than Zero, Signed | $if (Rr < 0) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Carry Flag Set | $if (C == 1) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Carry Flag Cleared | $if (C == 0) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Flag Set | $if (F == 1) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Flag Cleared | $if (F == 0) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Overflow Flag is Set | $if (OV == 1) PC = PC + k + 1$ | None | 1/2 |
| SEDI | k | Branch if Overflow Flag is Cleared | $if (OV == 0) PC = PC + k + 1$ | None | 1/2 |

Instruction Set Summary (Continued)

| Mnemonic | Operands | Description | Operation | Flags | #Cycles |
|--------------------------------------|----------|---|--|------------|---------|
| BRIS | R, R | Branch if Register Is Set | $R1 \leftarrow 1$ then PC \leftarrow PC + 4 + 1 | None | 1/2 |
| BRIS | R, R | Branch if Register Is Clear | $R1 \leftarrow 0$ then PC \leftarrow PC + 4 + 1 | None | 1/2 |
| DATA TRANSFER INSTRUCTIONS | | | | | |
| MOV | Rd, Rr | Move General Register | $Rd \leftarrow Rr$ | None | 1 |
| MOVW | Rd, Rr | Copy Register Word | $Rd \leftarrow Rr$ + 16 + 16 | None | 1 |
| LDI | Rd, K | Load Immediate | $Rd \leftarrow K$ | None | 1 |
| LDD | Rd, Rr | Load Direct | $Rd \leftarrow Rr$ | None | 2 |
| LDD | Rd, Rr | Load Indirect and Post-Inc | $Rd \leftarrow *Rr$; $Rr \leftarrow Rr + 1$ | None | 2 |
| LDD | Rd, Rr | Load Indirect and Pre-Dec | $Rr \leftarrow Rr - 1$; $Rd \leftarrow *Rr$ | None | 2 |
| LDD | Rd, Y | Load Indirect | $Rd \leftarrow *Y$ | None | 2 |
| LDD | Rd, Y+ | Load Indirect and Post-Inc | $Rd \leftarrow *Y$; $Y \leftarrow Y + 1$ | None | 2 |
| LDD | Rd, Y+ | Load Indirect and Pre-Dec | $Y \leftarrow Y - 1$; $Rd \leftarrow *Y$ | None | 2 |
| LDD | Rd, Y | Load Indirect with Displacement | $Rd \leftarrow Rr + \text{Displacement}$ | None | 2 |
| LDD | Rd, Z | Load Indirect | $Rd \leftarrow *Z$ | None | 2 |
| LDD | Rd, Z+ | Load Indirect and Post-Inc | $Rd \leftarrow *Z$; $Z \leftarrow Z + 1$ | None | 2 |
| LDD | Rd, Z+ | Load Indirect and Pre-Dec | $Z \leftarrow Z - 1$; $Rd \leftarrow *Z$ | None | 2 |
| LDD | Rd, X | Load Direct from SRAM | $Rd \leftarrow *X$ | None | 2 |
| ST | X, Rr | Store Indirect | $*X \leftarrow Rr$ | None | 2 |
| ST | X+, Rr | Store Indirect and Post-Inc | $*X \leftarrow Rr$; $X \leftarrow X + 1$ | None | 2 |
| ST | X-, Rr | Store Indirect and Pre-Dec | $X \leftarrow X - 1$; $*X \leftarrow Rr$ | None | 2 |
| ST | Y, Rr | Store Indirect | $*Y \leftarrow Rr$ | None | 2 |
| ST | Y+, Rr | Store Indirect and Post-Inc | $*Y \leftarrow Rr$; $Y \leftarrow Y + 1$ | None | 2 |
| ST | Y-, Rr | Store Indirect and Pre-Dec | $Y \leftarrow Y - 1$; $*Y \leftarrow Rr$ | None | 2 |
| STD | Y+, Rr | Store Indirect with Displacement | $*Y \leftarrow Rr + \text{Displacement}$ | None | 2 |
| ST | Z, Rr | Store Indirect | $*Z \leftarrow Rr$ | None | 2 |
| ST | Z+, Rr | Store Indirect and Post-Inc | $*Z \leftarrow Rr$; $Z \leftarrow Z + 1$ | None | 2 |
| ST | Z-, Rr | Store Indirect and Pre-Dec | $Z \leftarrow Z - 1$; $*Z \leftarrow Rr$ | None | 2 |
| STD | Z+, Rr | Store Indirect with Displacement | $*Z \leftarrow Rr + \text{Displacement}$ | None | 2 |
| STS | A, Rr | Store Direct to SRAM | $*X \leftarrow Rr$ | None | 2 |
| LPM | | Load Program Memory | $Rd \leftarrow *Z$ | None | 3 |
| LPM | Rd, Z | Load Program Memory | $Rd \leftarrow *Z$ | None | 3 |
| LPM | Rd, Z+ | Load Program Memory and Post-Inc | $Rd \leftarrow *Z$; $Z \leftarrow Z + 1$ | None | 3 |
| LPM | Rd, Z | Extended Load Program Memory | $Rd \leftarrow *RAMRZ$ | None | 3 |
| LPM | Rd, Z | Extended Load Program Memory | $Rd \leftarrow *RAMRZ$ | None | 3 |
| LPM | Rd, Z+ | Extended Load Program Memory and Post-Inc | $Rd \leftarrow *RAMRZ$; $Z \leftarrow Z + 1$ | None | 3 |
| SPM | | Store Program Memory | $*Z \leftarrow Rr$ | None | - |
| IN | Rd, P | In Port | $Rd \leftarrow P$ | None | 1 |
| OUT | P, Rr | Out Port | $P \leftarrow Rr$ | None | 1 |
| POP | Rr | Pop Register on Stack | $Rr \leftarrow *STACK$ | None | 2 |
| PUSH | Rr | Push Register on Stack | $Rr \leftarrow *STACK$ | None | 2 |
| BIT AND BIT-TEST INSTRUCTIONS | | | | | |
| SBIC | P, b | Set Bit in I/O Register | $IOIF(b) \leftarrow 1$ | None | 2 |
| SBIS | P, b | Clear Bit in I/O Register | $IOIF(b) \leftarrow 0$ | None | 2 |
| LSD | Rd | Logical Shift Left | $Rd(b+1) \leftarrow Rd(b)$; $Rd(0) \leftarrow 0$ | Z, C, N, V | 1 |
| LSD | Rd | Logical Shift Right | $Rd(b) \leftarrow Rd(b+1)$; $Rd(7) \leftarrow 0$ | Z, C, N, V | 1 |
| RCL | Rd | Rotate Left Through Carry | $Rd(b) \leftarrow C$; $Rd(b+1) \leftarrow Rd(b)$; $C \leftarrow Rd(7)$ | Z, C, N, V | 1 |
| RCR | Rd | Rotate Right Through Carry | $Rd(b) \leftarrow C$; $Rd(b+1) \leftarrow Rd(b)$; $C \leftarrow Rd(0)$ | Z, C, N, V | 1 |
| ASR | Rd | Arithmetic Shift Right | $Rd(b) \leftarrow Rd(b+1)$; $Rd(0) \leftarrow Rd(0)$ | Z, C, N, V | 1 |
| SWAP | Rd | Swap Nibbles | $Rd(3) \leftarrow Rd(7)$; $Rd(7) \leftarrow Rd(3)$ | None | 1 |
| SEI | s | Flag Set | $SREG \leftarrow 1$ | SREG | 1 |
| CLI | s | Flag Clear | $SREG \leftarrow 0$ | SREG | 1 |
| STC | Sr, b | Set Status from Register to T | $T \leftarrow SREG(b)$ | T | 1 |
| CLC | Sr, b | Clear Status from T to Register | $SREG(b) \leftarrow T$ | None | 1 |
| CLC | | Clear Carry | $C \leftarrow 0$ | C | 1 |
| SEN | | Set Negative Flag | $N \leftarrow 1$ | N | 1 |
| CLN | | Clear Negative Flag | $N \leftarrow 0$ | N | 1 |
| SEZ | | Set Zero Flag | $Z \leftarrow 1$ | Z | 1 |
| CLZ | | Clear Zero Flag | $Z \leftarrow 0$ | Z | 1 |
| SEI | | Global Interrupt Enable | $I \leftarrow 1$ | I | 1 |
| CLI | | Global Interrupt Disable | $I \leftarrow 0$ | I | 1 |
| SEIS | | Set Signed Test Flag | $S \leftarrow 1$ | S | 1 |
| CLIS | | Clear Signed Test Flag | $S \leftarrow 0$ | S | 1 |

Instruction Set Summary (Continued)

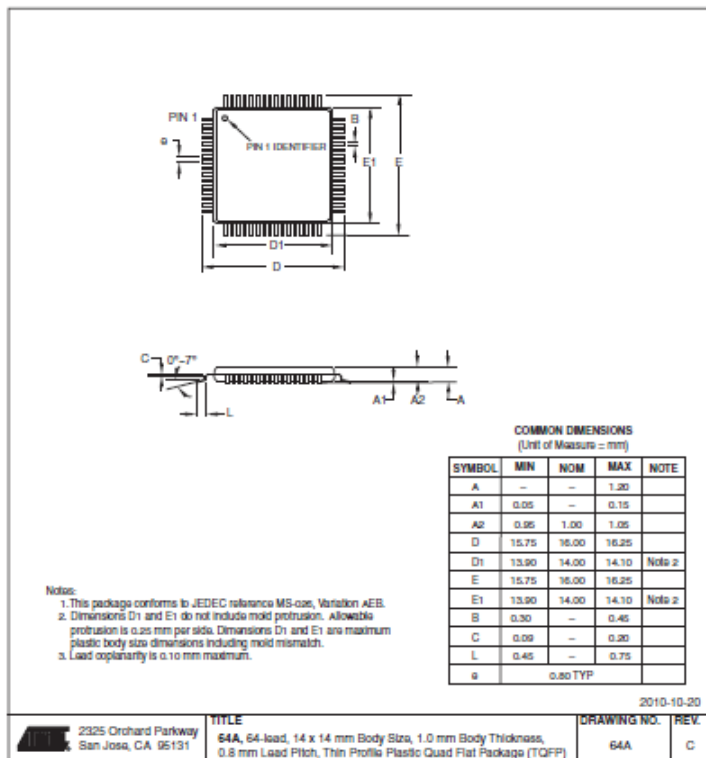
| Mnemonic | Operands | Description | Operation | Flags | #Cycles |
|---------------------------------|----------|--------------------------------|--|-------|---------|
| SEV | | Set Zero Complement Overflow | $Z \leftarrow 1$ | Z | 1 |
| CLV | | Clear Zero Complement Overflow | $Z \leftarrow 0$ | Z | 1 |
| SET | | Set T in SREG | $T \leftarrow 1$ | T | 1 |
| CLT | | Clear T in SREG | $T \leftarrow 0$ | T | 1 |
| SEI | | Set Interrupt Enable in SREG | $SI \leftarrow 1$ | SI | 1 |
| CLI | | Clear Interrupt Enable in SREG | $SI \leftarrow 0$ | SI | 1 |
| MCU CONTROL INSTRUCTIONS | | | | | |
| NOP | | No Operation | | None | 1 |
| SLEEP | | Sleep | (see specific device for Sleep function) | None | 1 |
| WDOR | | Watchdog Reset | (see specific device for WDR/Timer) | None | 1 |
| SEIAR | | Reset | For On-chip Debug Only | None | N/A |

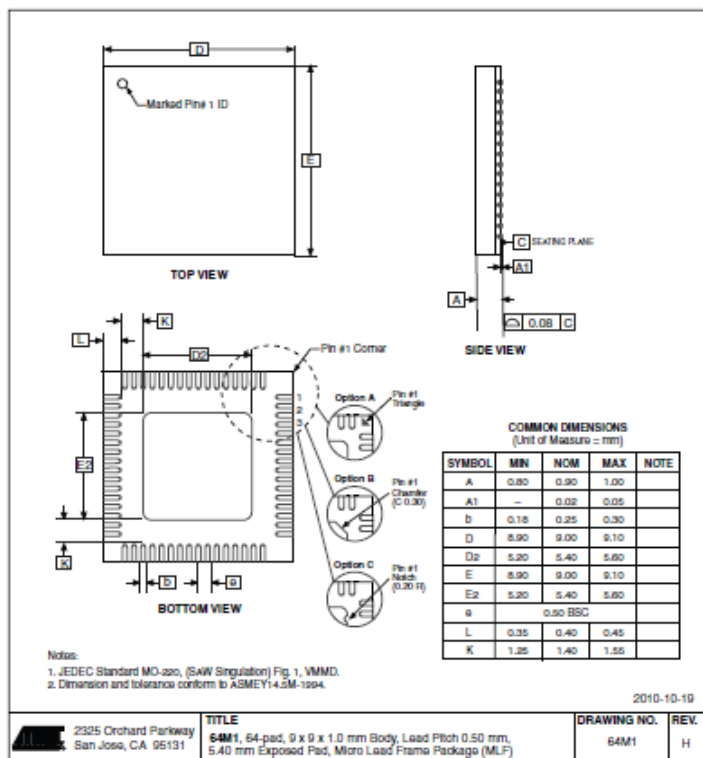
Ordering Information

| Speed (MHz) | Power Supply | Ordering Code ⁽¹⁾ | Package ⁽²⁾ | Operation Range |
|-------------|--------------|--------------------------------|------------------------|-------------------------------|
| 8 | 2.7 – 5.5V | ATmega128L-8AU | 64A | Industrial (-40°C to 85°C) |
| | | ATmega128L-8AUR ⁽³⁾ | 64A | |
| | | ATmega128L-8MU | 64M1 | |
| | | ATmega128L-8MUR ⁽³⁾ | 64M1 | |
| 16 | 4.5 – 5.5V | ATmega128-16AU | 64A | Industrial (-40°C to 85°C) |
| | | ATmega128-16AUR ⁽³⁾ | 64A | |
| | | ATmega128-16MU | 64M1 | |
| | | ATmega128-16MUR ⁽³⁾ | 64M1 | |
| 8 | 3.0 – 5.5V | ATmega128L-8AN | 64A | Extended (-40°C to 105°C) |
| | | ATmega128L-8ANR ⁽³⁾ | 64A | |
| | | ATmega128L-8MN | 64M1 | |
| | | ATmega128L-8MNR ⁽³⁾ | 64M1 | |
| 16 | 4.5 – 5.5V | ATmega128-16AN | 64A | Extended (-40°C to 105°C) |
| | | ATmega128-16ANR ⁽³⁾ | 64A | |
| | | ATmega128-16MN | 64M1 | |
| | | ATmega128-16MNR ⁽³⁾ | 64M1 | |

- Notes:
1. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 2. The device can also be supplied in water form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 3. Tape and Reel

| Package Type | |
|--------------|---|
| 64A | 64-lead, 14 x 14 x 1.0mm, Thin Profile Plastic Quad Flat Package (TQFP) |
| 64M1 | 64-pad, 9 x 9 x 1.0mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF) |

Packaging Information
64A


64M1


LAMPIRAN B
DATA SHEET
Proximity Hall Effect sensor NJK500-2A



Gambar B. *Proximity Hall Effect sensor NJK500-2A*

Aplikasi:

1. Mendeteksi kecepatan putaran (Tachometer)
2. Sensor deteksi efek hall

Spesifikasi Proximity Hall Effect sensor NJK500-2A:

- | | |
|------------------------------|--------------------------|
| > Tipe output | : NPN wire normally open |
| > Rentang deteksi | : 10 mm |
| > Tegangan catu daya | : 5 - 24VDC |
| > Arus keluaran | : 200mA |
| > Objek yang dapat dideteksi | : magnet |
| > Mengalihkan frekuensi | : 320 KHz |
| > Shell material | : tembaga |

Pin Wiring:

Kabel coklat - Vcc

Kabel biru - GND

Kabel hitam - Data

LAMPIRAN C LISTING PROGRAM

```
/*  
*
```

This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :
Version :
Date : 07/07/2018
Author : tyery08
Company : embeeminded.blogspot.com
Comments:

Chip type : ATmega128A
Program type : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 1024

```
*****  
/  

```

```
#include <mega128a.h>  
#include <alcd.h>  
#include <stdlib.h>  
#include <delay.h>  
#include <stdio.h>  
#include <i2c.h>  
#include <ds1307.h>
```

```
#define pompa1 PORTD.4
#define pompa2 PORTD.5
#define mov1 PORTB.6
#define mov2 PORTB.7
#define b3 PORTA.0
#define b2 PORTA.2
#define b1 PORTA.4
#define b0 PORTA.6
```

```
char temp1 [16];
int waktu = 0,persen;
unsigned char Rpm, Rpm2;
float jrk,acuan_mov; //,acuan_mov_min;
int jarak,count=0,drajat=0,x,y;
char buff[32], buff3[50], buff4[50];
float freq,volt,adc,adc_arus; // to store value of frequency value
unsigned char i=0,dur;
int tegangan;
float av_adc,av_adc1,vout,arus,daya;
char buffer[32], buffer1[10]; // to store the frequency value as a
string to be displayed on lcd
int counter=0,flag=0;
int flow;
float flow_rate;
int max_step=150;
int hitungstep;
char temp[16], temp1[16];
unsigned char wd;
unsigned char dd;
unsigned char mm;
unsigned char yy;
unsigned char s;
unsigned char m;
unsigned char h;
```

```
// External Interrupt 4 service routine
```

```
interrupt [EXT_INT4] void ext_int4_isr(void)
{
// Place your code here
    counter++;
//counter1++;
}
```

```
#ifndef RXB8
#define RXB8 1
#endif
```

```
#ifndef TXB8
#define TXB8 0
#endif
```

```
#ifndef UPE
#define UPE 2
#endif
```

```
#ifndef DOR
#define DOR 3
#endif
```

```
#ifndef FE
#define FE 4
#endif
```

```
#ifndef UDRE
#define UDRE 5
#endif
```

```
#ifndef RXC
#define RXC 7
#endif
```

```
#define FRAMING_ERROR (1<<FE)
```

```

#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// Get a character from the USART1 Receiver
#pragma used+
char getchar1(void)
{
    char status,data;
    while (1)
    {
        while (((status=UCSR1A) & RX_COMPLETE)==0);
        data=UDR1;
        if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
            return data;
    }
}
#pragma used-

// Write a character to the USART1 Transmitter
#pragma used+
void putchar1(char c)
{
    while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
    UDR1=c;
}
#pragma used-

#define _ALTERNATE_PUTCHAR_
#include <stdio.h>
#define USART0 0           // agar pembacaan tidak acak
#define USART1 1
unsigned char poutput;

```

```

void putchar(char c)
{
switch (poutput)
{
case USART0: // the output will be directed to USART0
while ((UCSR0A & DATA_REGISTER_EMPTY)==0);
UDR0=c;
break;

case USART1: // the output will be directed to USART1
while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
UDR1=c;
break;
};
}

```

// Standard Input/Output functions

```
#include <stdio.h>
```

// Timer 0 overflow interrupt service routine

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
```

```
{
```

// Reinitialize Timer 0 value

```
TCNT0=0x8A;
```

// Place your code here

```
    waktu++;
```

```
    if (waktu >=132){//kira2 bisa mencapai 1 detik
```

```
//detik++;
```

```
Rpm = counter * 60.0;
```

```
Rpm2 = Rpm * 4.0;
```

```
counter = 0;
```

```
waktu = 0;
```

```
Rpm2 = Rpm2;
```

```
}
```

```
}
```

```

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Place your code here
    i++;
}

#define ADC_VREF_TYPE 0x60

// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}

// Declare your global variables here
void baca_volt(){
    for(i=0;i<10;i++){
        adc=read_adc(0);
        av_adc=av_adc+adc;
    }
    av_adc=av_adc/10;
    volt=av_adc*106/255;

}

void baca_arus (){
    for (i=0;i<10;i++){

```

```

adc_arus=read_adc(1);
av_adc1=av_adc1+adc_arus;
}
av_adc1=av_adc1/10;
vout=av_adc1*106/255;
arus=fabs(vout-2.50)/0.066/1000;
daya=arus*volt;
}
void baca_ultrasonic(){
    count=0;
    PORTA.1=1;
    delay_us(15);
    PORTA.1=0;
    while(PINA.3==0){ };
    while(PINA.3==1){ count++; };
    jrk=count;
    jrk=jrk/100;
    jrk=65.5-jrk;
    jarak = jrk;
    /*if (jrk<0){
        lcd_gotoxy(0,0);
        lcd_putsf("ERROR");}
    else if (jrk>70)
    { lcd_gotoxy(0,0);
        lcd_putsf("ERROR");}
    else {
        lcd_gotoxy(0,0);
        lcd_putsf("Lvl:");
        //sprintf(buffer,"%d ",jrk);
        ftoa(jrk,2,buffer);
        lcd_puts(buffer);

        lcd_gotoxy(10,0);
        lcd_putsf("Lvl:");
        sprintf(buffer,"%d ",jarak);
        lcd_puts(buffer);

```



```

    lcd_putsf(" Cm ");
    delay_ms(100);}*/
}
void baca_flow(){
    TIMSK=0x05;
    TCCR1B=0x07;
    delay_ms(100);
    TCCR1B=0x00;
    TIMSK=0x00;
    dur=TCNT1;
    //flow_rate=0;
    freq = (((dur + i*65536)*600)/4.8)*0.0166;
    TCNT1=0x0000;
    i=0;
    flow_rate=freq;
    flow_rate=flow_rate-1359872.00;
    flow = flow_rate;
    lcd_gotoxy(0,2);
    lcd_putsf("F:");
    //sprintf(buffer,"%d",flow_rate-18350);
    ftoa(flow_rate,2,buffer);
    lcd_puts(buffer);
    lcd_putsf("L/M ");
}
void cecewe_start(){
    b0=0; b1=0; b2=0; b3=1;
    delay_ms(5);
    b0=0; b1=0; b2=1; b3=0;
    delay_ms(5);
    b0=0; b1=1; b2=0; b3=0;
    delay_ms(5);
    b0=1; b1=0; b2=0; b3=0;
    delay_ms(5);
    b0=0; b1=0; b2=0; b3=1;
    delay_ms(5);
    b0=0; b1=0; b2=1; b3=0;

```

```

    delay_ms(5);
    b0=0; b1=1; b2=0; b3=0;
    delay_ms(5);
    b0=1; b1=0; b2=0; b3=0;
    delay_ms(5);
}

void cecewe(){ //close
    b0=0; b1=0; b2=0; b3=1;
    delay_ms(5);
    b0=0; b1=0; b2=1; b3=0;
    delay_ms(5);
    b0=0; b1=1; b2=0; b3=0;
    delay_ms(5);
    b0=1; b1=0; b2=0; b3=0;
    delay_ms(5);
    b0=0; b1=0; b2=0; b3=1;
    delay_ms(5);
    b0=0; b1=0; b2=1; b3=0;
    delay_ms(5);
    b0=0; b1=1; b2=0; b3=0;
    delay_ms(5);
    b0=1; b1=0; b2=0; b3=0;
    delay_ms(5);
    hitungstep--;
    persen=(hitungstep*100)/max_step;
    if(persen<0){
        persen=0;
    }
    /* lcd_gotoxy(13,1);
    sprintf(buffer,"%d ",persen);
    lcd_puts(buffer); */

}

void cewe(){ //open
    b0=1; b1=0; b2=0; b3=0;

```

```

delay_ms(5);
b0=0; b1=1; b2=0; b3=0;
delay_ms(5);
b0=0; b1=0; b2=1; b3=0;
delay_ms(5);
b0=0; b1=0; b2=0; b3=1;
delay_ms(5);
b0=1; b1=0; b2=0; b3=0;
delay_ms(5);
b0=0; b1=1; b2=0; b3=0;
delay_ms(5);
b0=0; b1=0; b2=1; b3=0;
delay_ms(5);
b0=0; b1=0; b2=0; b3=1;
delay_ms(5);
hitungstep++;
persen=(hitungstep*100)/max_step;
if(persen>100){
    persen=100;
}
/* lcd_gotoxy(13,1);
sprintf(buffer,"%d ",persen);
lcd_puts(buffer);*/
}
void stopped(){
    b3=0; b2=0; b1=0; b0=0;
}
void tampil_persen(){

}

void mov(int kondisi){
    if(kondisi == 1){ //MOV CLOSE
        mov1 = 1;
        mov2 = 0;
        drajat -= 50;
    }
}

```

```

    delay_ms(300);
}
if(kondisi == 2){ //MOV OPEN
    mov1 = 0;
    mov2 = 1;
    drajat += 50;
    delay_ms(2700);
}
if(kondisi == 3){ //MOV no operation
    mov1 = 1;
    mov2 = 1;
} }

```

```

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTA.1=0;
PORTA.3=1;
DDRA.0=1;
DDRA.1=1;
DDRA.2=1;
DDRA.3=0;
DDRA.4=1;
DDRA.6=1;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In

```

```
// State7=T State6=T State5=T State4=T State3=T State2=T  
State1=T State0=T  
PORTB=0x00;  
DDRB=0xC0;
```

```
// Port C initialization  
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In  
Func1=In Func0=In  
// State7=T State6=T State5=T State4=T State3=T State2=T  
State1=T State0=T  
PORTC=0x00;  
DDRC=0x00;
```

```
// Port D initialization  
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In  
Func1=In Func0=In  
// State7=T State6=T State5=T State4=T State3=T State2=T  
State1=T State0=T  
PORTD=0x00;  
DDRD.4=1;  
DDRD.5=1;
```

```
// Port E initialization  
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In  
Func1=In Func0=In  
// State7=T State6=T State5=T State4=T State3=T State2=T  
State1=T State0=T  
PORTE=0x00;  
DDRE=0x00;
```

```
// Port F initialization  
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In  
Func1=In Func0=In  
// State7=T State6=T State5=T State4=T State3=T State2=T  
State1=T State0=T  
PORTF=0x00;
```

```
DDRF=0x00;
```

```
// Port G initialization
```

```
// Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State4=T State3=T State2=T State1=T State0=T
```

```
PORTG=0x00;
```

```
DDRG=0x00;
```

```
// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: 250,000 kHz
```

```
// Mode: Normal top=0xFF
```

```
// OC0 output: Disconnected
```

```
ASSR=0x00;
```

```
TCCR0=0x07;
```

```
TCNT0=0x8A;
```

```
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: 16000,000 kHz
```

```
// Mode: Normal top=0xFFFF
```

```
// OC1A output: Discon.
```

```
// OC1B output: Discon.
```

```
// OC1C output: Discon.
```

```
// Noise Canceler: Off
```

```
// Input Capture on Falling Edge
```

```
// Timer1 Overflow Interrupt: On
```

```
// Input Capture Interrupt: Off
```

```
// Compare A Match Interrupt: Off
```

```
// Compare B Match Interrupt: Off
```

```
// Compare C Match Interrupt: Off
```

```
TCCR1A=0x00;
```

```
TCCR1B=0x01;
```

```
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```

```
ICR1H=0x00;  
ICR1L=0x00;  
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;  
OCR1CH=0x00;  
OCR1CL=0x00;
```

```
// Timer/Counter 2 initialization  
// Clock source: System Clock  
// Clock value: Timer2 Stopped  
// Mode: Normal top=0xFF  
// OC2 output: Disconnected  
TCCR2=0x00;  
TCNT2=0x00;  
OCR2=0x00;
```

```
// Timer/Counter 3 initialization  
// Clock source: System Clock  
// Clock value: Timer3 Stopped  
// Mode: Normal top=0xFFFF  
// OC3A output: Discon.  
// OC3B output: Discon.  
// OC3C output: Discon.  
// Noise Canceler: Off  
// Input Capture on Falling Edge  
// Timer3 Overflow Interrupt: Off  
// Input Capture Interrupt: Off  
// Compare A Match Interrupt: Off  
// Compare B Match Interrupt: Off  
// Compare C Match Interrupt: Off  
TCCR3A=0x00;  
TCCR3B=0x00;  
TCNT3H=0x00;  
TCNT3L=0x00;
```

```
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;
```

```
// External Interrupt(s) initialization
```

```
// INT0: Off
```

```
// INT1: Off
```

```
// INT2: Off
```

```
// INT3: Off
```

```
// INT4: On
```

```
// INT4 Mode: Rising Edge
```

```
// INT5: Off
```

```
// INT6: Off
```

```
// INT7: Off
```

```
EICRA=0x00;
```

```
EICRB=0x03;
```

```
EIMSK=0x10;
```

```
EIFR=0x10;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x05;
```

```
ETIMSK=0x00;
```

```
// USART0 initialization
```

```
// Communication Parameters: 8 Data, 1 Stop, No Parity
```

```
// USART0 Receiver: On
```

```
// USART0 Transmitter: On
```

```
// USART0 Mode: Asynchronous
```

```
// USART0 Baud Rate: 9600
```

```
UCSR0A=0x00;
```



```
UCSR0B=0x98;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x67;

// USART1 initialization
UCSR1A=0x00;
UCSR1B=0x18;
UCSR1C=0x06;
UBRR1H=0x00;
UBRR1L=0x67;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 1000,000 kHz
// ADC Voltage Reference: AVCC pin
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

i2c_init();
rtc_init(0,0,0);
```

```

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD
menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 3
// D5 - PORTC Bit 4
// D6 - PORTC Bit 5
// D7 - PORTC Bit 6
// Characters/line: 20
lcd_init(20);

// Global enable interrupts
#asm("sei")

/*****setup time and date*****/
h=22;m=28;s=00;
rtc_set_time(h,m,s);

dd=16;mm=07;yy=18;
rtc_set_date(wd,dd,mm,yy);
/*****/

pompa1=1;
pompa2=0;
mov1=0;
mov2=1;
lcd_gotoxy(0,0);
lcd_putsf("  WELCOME TO  ");
lcd_gotoxy(0,1);
lcd_putsf("  PLTMH PLANT  ");
lcd_gotoxy(0,2);
lcd_putsf("    2018    ");

```

```

for(i=0;i<160;i++){
    cecewe_start();
}

delay_ms(1500);
lcd_clear();
while (1)
{
    // Place your code here
    start:
    baca_ultrasonic();
    baca_flow();
    baca_volt();
    baca_arus();

    lcd_gotoxy(0,0);
    lcd_putsf("L:");
    sprintf(buff,"L:%d cm ",jarak);
    ftoa(jrk,2,buff);
    lcd_puts(buff);
    lcd_putsf("cm");

    lcd_gotoxy(11,0);
    lcd_putsf("V:");
    ftoa(volt,2,buff);
    lcd_puts(buff);
    lcd_putsf("V");

    sprintf(temp1,"N:%d RPM", Rpm);
    lcd_gotoxy(0,1);
    lcd_puts(temp1);

    lcd_gotoxy(11,1);
    lcd_putsf("I:");
    ftoa(arus,2,buff);
    lcd_puts(buff);

```

```

lcd_putsf("A");

lcd_gotoxy(11,2);
lcd_putsf("P:");
ftoa(daya,2,buff);
lcd_puts(buff);
lcd_putsf("W");

lcd_gotoxy(11,3);
sprintf(buffer,"MOV2:%d",persen);
lcd_puts(buffer);
lcd_putsf("%");

lcd_gotoxy(0,3);
sprintf(buffer1,"N2:%d Rpm",Rpm2);
lcd_puts(buffer1);
lcd_putsf("%");

//=====data logger=====
sprintf(buff3, "%02u:%02u:%02u/%02u:%02u:%02u ; %.2f
cm ; %d RPM ; %.2f lps ; %.2f V\r",dd,mm,yy,s,m,h, jrk, Rpm,
flow_rate, volt);
poutput = USART1;
puts(buff3);

//=====
===HMI=====
==

// matikan salah satu kodingan, karena dalam kodingan ini
kedua HMI belum diintegrasikan

//=====HMI delima=====
// sprintf(buff4, "%.2f;%d\r", volt, Rpm);
// poutput = USART0;
// puts(buff4);

```

```

    poutput = USART0;
    printf("%.2f, %d, %d, %.2f, %.2f \r", jrk, Rpm, Rpm2,
flow_rate, volt);

```

```

    /*//=====HMI Ragil=====
    printf ("JARAK: %.2f;", jrk);
    printf ("RPM: %d;", Rpm);
    printf ("FLOW: %.2f;", flow_rate);
    printf ("TEGANGAN: %.2f; \n", volt);
    poutput = USART0;
    puts(buff4);*/
    /*sprintf(buff4, "JARAK: %.2f; RPM: %d; FLOW: %.2f;
TEGANGAN: %.2f\r", jrk, Rpm, flow_rate, volt);
    poutput = USART0;
    puts(buff4);*/

```

```

//=====
=====
==

```

```

    // delay_ms(300);

```

```

    if(volt<30){
        ceuwe();
        tampil_persen();
    }
    else if(volt>30.5){
        ceceuwe();
        tampil_persen();
    }
    else {
        stopped();
        tampil_persen();
    }
    acuan_mov= jrk - 35;

```

```

        if(jarak==35 && flag==0){
pompa1 =0; //pompa1
pompa2 =0; //pompa2
mov1=0;
delay_ms(1500);
mov1=1;
flag=1;
delay_ms(1000);
goto start;
    }

if(jarak>45 && flag==1){
    pompa1=1;
    pompa2=1;

}
else if(jarak>35 && jarak<45 && flag==1){
    pompa1=0;
    pompa2=0;
    mov2=0; //
    mov1=1;
    drajat=0;
    x=1;
    y=0;
}
else if(jarak==35 && flag==1 && x==1){
    pompa1=0;
    pompa2=0;
    if(drajat < 50){
        mov(2); //MOV OPEN
        mov(3);
    }
}
else if(jarak==35 && flag==1&& y==1){
    pompa1=0;
    pompa2=0;

```

```

if(drajat > 50){
    mov(1);    //MOV OPEN
    mov(3);

}
}
else if(jarak>25 && jarak<35 && flag==1){
    pompa1=0;
    pompa2=0;
    mov1=1; //
    mov2=0;
    drajat=100;
    x=1;
    y=0;
}
else if(jarak<25 && flag==1){
    pompa1=1;
    pompa2=1;

} //delay_ms(250);
}
}

```

LAMPIRAN D
DATA KALIBRASI SENSOR TEGANGAN

| Kec. Turbin | Pembacaan validator(X) | Pembacaan Alat | | Rata-Rata | Koreksi (Y) | H(I) | H(I)/X |
|-------------|------------------------|----------------|-----------------|-----------|-------------|--------|--------------|
| | | Rata-rata Naik | Rata-rata Turun | | | | |
| 360 | 3 | 2.788 | 2.774 | 2.781 | 0.226 | -0.014 | -0.004666667 |
| 475 | 5 | 4.878 | 5.038 | 4.958 | -0.038 | 0.16 | 0.032 |
| 680 | 7 | 6.998 | 7.056 | 7.027 | -0.056 | 0.058 | 0.008285714 |
| 797 | 9 | 9.01 | 9.024 | 9.017 | -0.024 | 0.014 | 0.001555556 |
| 800 | 11 | 11.022 | 11.068 | 11.045 | -0.068 | 0.046 | 0.004181818 |
| 938 | 13 | 13.016 | 12.994 | 13.005 | 0.006 | -0.022 | -0.001692308 |
| 1032 | 15 | 15.354 | 15.122 | 15.238 | -0.122 | -0.232 | -0.015466667 |
| 1073 | 17 | 16.986 | 17.022 | 17.004 | -0.022 | 0.036 | 0.002117647 |
| 1040 | 19 | 19.036 | 19.03 | 19.033 | -0.03 | -0.006 | -0.000315789 |
| 1057 | 21 | 20.992 | 21.04 | 21.016 | -0.04 | 0.048 | 0.002285714 |
| 1057 | 23 | 23.018 | 23.102 | 23.06 | -0.102 | 0.084 | 0.003652174 |
| Jumlah | | | | | -0.168 | 0.088 | 0.028285019 |
| Rata-Rata | | | | | -0.0168 | 0.0088 | 0.002828502 |

| No | X | X ² | Y-Y' | X*Y | (Y-Y') ² | a | b | Yreg | (Y-Yreg) ² |
|-----------|-----|----------------|--------------|---------|---------------------|----------|--------------|--------------|-----------------------|
| 1 | 3 | 9 | 0.2428 | 0.678 | 0.05895184 | 0.050144 | -0.005578698 | 0.033408284 | 0.037091569 |
| 2 | 5 | 25 | -0.0212 | -0.19 | 0.00044944 | | | 0.022250888 | 0.003630169 |
| 3 | 7 | 49 | -0.0392 | -0.392 | 0.00153664 | | | 0.011093491 | 0.004501537 |
| 4 | 9 | 81 | -0.0072 | -0.216 | 5.184E-05 | | | -6.39053E-05 | 0.000572937 |
| 5 | 11 | 121 | -0.0512 | -0.748 | 0.00262144 | | | -0.011221302 | 0.003223821 |
| 6 | 13 | 169 | 0.0228 | 0.078 | 0.00051984 | | | -0.022378698 | 0.000805351 |
| 7 | 15 | 225 | -0.1052 | -1.83 | 0.01106704 | | | -0.033536095 | 0.007825863 |
| 8 | 17 | 289 | -0.0052 | -0.374 | 2.704E-05 | | | -0.044693491 | 0.000514995 |
| 9 | 19 | 361 | -0.0132 | -0.57 | 0.00017424 | | | -0.055850888 | 0.000668268 |
| 10 | 21 | 441 | -0.0232 | -0.84 | 0.00053824 | | | -0.067008284 | 0.000729447 |
| 11 | 23 | 529 | -0.102 | -2.346 | 0.010404 | | | -0.07816568 | 0.000568075 |
| Jumlah | 120 | 1770 | -3.46945E-17 | -4.404 | 0.0759376 | | | SSR | 0.059563956 |
| Rata-Rata | 12 | 177 | -3.46945E-18 | -0.4404 | 0.00759376 | | | | |

LAMPIRAN E
DATA KALIBRASI SENSOR PROXIMITY HALL EFFECT

| Flow | Pembacaan Validator (X) | Pembacaan Alat | | Rata-Rata | Koreksi (Y) | H(I) | H(I)/X |
|-----------|-------------------------|----------------|-----------------|-----------|-------------|----------|--------------|
| | | Rata-rata Naik | Rata-rata Turun | | | | |
| 26.75 | 12 | 12.48 | 12.68 | 12.58 | -0.68 | 0.2 | 0.016666667 |
| 26.9 | 60 | 63.8 | 67.3 | 65.55 | -7.3 | 3.5 | 0.058333333 |
| 26.75 | 104 | 94.46 | 94.46 | 94.46 | 9.54 | 0 | 0 |
| 27.14 | 148 | 149.36 | 149.46 | 149.41 | -1.46 | 0.1 | 0.000675676 |
| 27.38 | 180 | 168.8 | 168.8 | 168.8 | 11.2 | 0 | 0 |
| 27.62 | 196 | 201.6 | 201.8 | 201.7 | -5.8 | 0.2 | 0.001020408 |
| 27.86 | 208 | 204 | 202.8 | 203.4 | 5.2 | -1.2 | -0.005769231 |
| 28.1 | 240 | 234.4 | 234.4 | 234.4 | 5.6 | 0 | 0 |
| 28.87 | 252 | 250 | 246.4 | 248.2 | 5.6 | -3.6 | -0.014285714 |
| Jumlah | | | | | 21.9 | -0.8 | 0.056641139 |
| Rata-Rata | | | | | 2.433333333 | -0.08889 | 0.00629346 |

| No | X | X ² | Y-Y' | X*Y | (Y-Y')^2 | a | b | Yreg | (Y-Yreg)^2 |
|-----------|-------------|----------------|--------------|-------------|-------------|----------|-------------|--------------|-------------|
| 1 | 12 | 144 | -3.113333333 | -8.16 | 9.692844444 | -1.32573 | 0.024165396 | -1.035743479 | 0.126553423 |
| 2 | 60 | 3600 | -9.733333333 | -438 | 94.73777778 | | | 0.124195517 | 55.11867908 |
| 3 | 104 | 10816 | 7.106666667 | 992.16 | 50.50471111 | | | 1.18747293 | 69.76470845 |
| 4 | 148 | 21904 | -3.893333333 | -216.08 | 15.15804444 | | | 2.250750343 | 13.76966811 |
| 5 | 180 | 32400 | 8.766666667 | 2016 | 76.85444444 | | | 3.024043007 | 66.84627275 |
| 6 | 196 | 38416 | -8.233333333 | -1136.8 | 67.78777778 | | | 3.410689339 | 84.8367981 |
| 7 | 208 | 43264 | 2.766666667 | 1081.6 | 7.654444444 | | | 3.700674088 | 2.24797819 |
| 8 | 240 | 57600 | 3.166666667 | 1344 | 10.02777778 | | | 4.473966752 | 1.267950875 |
| 9 | 252 | 63504 | 3.166666667 | 1411.2 | 10.02777778 | | | 4.763951501 | 0.698977093 |
| Jumlah | 1400 | 271648 | 0 | 5045.92 | 342.4456 | | | SSR | 294.6775861 |
| Rata-Rata | 155.5555556 | 30183.11111 | 0 | 560.6577778 | 38.04951111 | | | | |

LAMPIRAN F
DATA SHEET
SENSOR TEGANGAN



Gambar F. Sensor Tegangan

Sensor tegangan di atas dibuat menggunakan prinsip pembagi tegangan.

- R1 : 827k Ω
- R2 : 43k Ω
- Vinput : 0 – 100 VDC
- Voutput : 0 – 5 VDC

BIODATA PENULIS



Penulis merupakan anak pertama dari dua bersaudara yang dilahirkan di Lumajang pada 17 Oktober 1997 dari pasangan Sudarsono dan Heny Lailiyah Mariana. Se-masa kecil penulis telah menempuh pendidikan formal di SDN Kedungrejo 01, SMPN 1 Yosowilangun dan SMAN 2 Lumajang. Pada pertengahan tahun 2015 penulis diterima di Jurusan D3 Teknologi Instrumentasi Fakultas Vokasi ITS, melalui jalur SMITS dan terdaftar sebagai mahasiswa dengan NRP 10511500000005. Selama perkuliahan, penulis aktif dalam berbagai kegiatan dan organisasi mahasiswa. Beberapa organisasi yang sempat ditekuni penulis yaitu staff di BKK JMMI ITS (2016-2017), sebagai mentor di LDJ Fusi Ulul Albab (2016-2017), sebagai staff Kemuslimahan di LDJ Fusi Ulul Albab (2016-2017), sebagai Bendahara Umum di Ikatan Mahasiswa Muhammadiyah (2016-2017), sebagai staff Kemuslimahan di LDJ Uwais Alqarni (2017-2018), sebagai Bendahara Umum di Ikatan Mahasiswa Muhammadiyah (2017-2018), dan sebagai Sekretaris Bidang Keilmuan di Ikatan Mahasiswa Muhammadiyah (2017-2018). Dalam hal akademik, penulis pernah memegang peranan penting sebagai asisten Praktikum Fisika Terapan dan asisten praktikum Rangkaian Listrik. Akhir kata bila ada kritik dan saran dapat menghubungi penulis melalui : delimapalwa@gmail.com